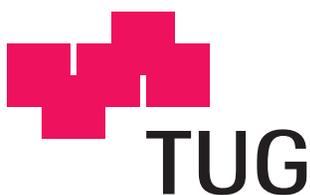


Diplomarbeit

Messgerät zur Erfassung relevanter Parameter im Stromversorgungsnetz mit Webinterface

Michitsch Ingo

Institut für Elektronik
Technische Universität Graz
Vorstand: Univ.-Prof. Dipl.-Ing. Dr. techn. Wolfgang Pribyl



Begutachter: Ass.-Prof. Dipl.-Ing. Dr. techn. Gunter Winkler

Graz, im November 2009

Eidesstattliche Erklärung

Ich erkläre ehrenwörtlich, dass ich die vorliegende Arbeit selbstständig und ohne fremde Hilfe verfasst, andere als die angegebenen Quellen nicht benutzt und die den Quellen wörtlich oder inhaltlich entnommenen Stellen als solche kenntlich gemacht habe.

Ort, Datum

Unterschrift
Name des / der Studierenden

Danksagung

An dieser Stelle möchte ich allen Personen danken, die mir während meiner Diplomarbeit mit Rat und Tat zur Seite standen.

Für die Betreuung von universitärer Seite bedanke ich mich bei Herrn Ass.-Prof. Dipl.-Ing. Dr. techn. Gunter Winkler.

Für die Unterstützung von außeruniversitärer Seite bedanke ich mich bei Herrn Dipl.-Ing. Franz Blasge und der Firma KNAPP.

Schlussendlich gilt mein Dank meiner Familie sowie meiner Freundin, für die moralische und materielle Unterstützung während meiner gesamten Ausbildungszeit.

Graz, im November 2009

Michitsch Ingo



Kurzfassung

Das Ziel dieser Arbeit ist die Entwicklung eines Messgerätes zur Erfassung der relevanten Messgrößen eines dreiphasigen Stromversorgungsnetzes und der Weiterleitung der gesammelten Daten nach dem Vorbild eines modernen Smart Meter. Der Einsatzzweck soll dabei die Überwachung elektrischer Verbraucher vor Ort sein, sowie die Übertragung der anfallenden Messdaten über ein Netzwerk als auch deren Aufzeichnung auf einem USB-Flash Speichermedium. Als relevante Parameter sind die Effektivwerte von Spannung und Strom, ebenso wie Schein-, Wirk-, Blindenergie, Leistung, Powerfaktor und Netzfrequenz der einzelnen Phasen zu verstehen. Ferner erfolgt auch die Bestimmung der Energieflussrichtung zur Feststellung, ob dem Netz Energie entzogen oder eingespeist wird.

Mittels eines USB-Controllers wird das Messgerät zu einem Datenlogger mit einer Schnittstelle für ein externes Flash Speichermedium erweitert, sodass darauf die gewonnenen Messwerte im Dateiformat FAT, als CSV Datei im wählbaren Intervall abgespeichert werden können. Diese tabellarisch organisierte Datei kann am Computer beispielsweise mit einem Tabellenkalkulationsprogramm betrachtet und die aufgezeichneten Messdaten visualisiert werden.

Über einen Embedded Web Server ist sowohl der Fernzugriff, als auch die Datenübertragung über ein Netzwerk und das Internet möglich. Um unabhängig von einer bestimmten Plattform zu sein, wird vom Browser bei Abfrage des Servers ein Java Applet geladen, das als Programm mit einer grafischen Oberfläche dem Benutzer dynamisch die momentanen Messgrößen darbietet. Die übertragenen Messwerte können dabei zusätzlich auch grafisch, als Funktion der Zeit dargestellt werden.

Abstract

The aim of this work is the development of a measurement device for the acquisition of notable measurement parameters on a three-phase grid and the transfer of the collected data along the lines of an advanced smart meter. The operation purpose is the on-the-spot monitoring of electric consumers, the transmission of the gathered data via network and their storage on an USB flash memory device. The term notable measurement parameters refers not only to RMS values of voltage and current but also to apparent energy, active energy and reactive energy as well as to power, power factor and frequency of individual phases. Furthermore, the energy flow direction will be determined in order to discover whether energy is consumed from the grid or placed back to it.

An additional USB controller provides the functionality to record data on a flash memory device at adjustable intervals. The file system, which is in this case FAT, must be organized to maintain the physical location of the file in which the data is collected. Because of the structure of the data, the CSV file format is appropriate for this task. This file is arranged as a simple spreadsheet and can therefore be viewed and edited, for example, through spreadsheet software.

By the use of an embedded web server, remote access to the device and the data through a network connection or over the Internet is possible. In order to be independent of a certain computer system, a Java applet that is integrated on an HTML site, located on the server, is loaded and executed. Java applet provides a graphical user interface and a dynamic display of present measurement parameters. Additionally, the transferred measurement data can be graphically displayed as a function of time.

Inhaltsverzeichnis

1	Einleitung	13
1.1	Motivation	13
1.2	Aufgabenstellung	14
1.3	Entwurf und Umsetzung	15
2	Mikrocontroller	16
2.1	XC167	16
2.1.1	RTC	20
3	Mess-Signalaufnahme und Verarbeitung	22
3.1	ADE7758	23
3.1.1	Analog-Eingänge	24
3.1.2	Effektivwertberechnung	27
3.1.3	Wirkleistung	28
3.1.4	Blindleistung	31
3.1.5	Scheinleistung	32
3.1.6	Periodendauermessung/Frequenzmessung	33
3.1.7	Power Faktor	33
3.1.8	Takt	35
3.1.9	Interrupts	36
3.1.10	SPI Schnittstelle	36
3.2	Anpassungsnetzwerk und Präzisionsspannungsteiler	39
3.3	Antialiasing Filter	42
3.4	Messbereich	43
3.5	Kalibrierung und Justierung	44
3.5.1	Kalibriergrundlage	45
3.5.2	Beschreibung Umsetzerkennlinie	46
3.5.3	Effektivwerte	46
3.5.4	Regressionsrechnung MKQ	48
3.5.5	Intervallweise lineare Approximation	49
3.5.6	Phasenkalibrierung	49
3.5.7	Energie bzw. Leistung	50
3.5.8	Power Faktor	54
3.6	Schaltungen zur Leistungsmessung	55

4	Netzwerk und Internetanbindung	58
4.1	OSI und Internet Schichtmodell	58
4.2	Hardware	60
4.3	Ethernet	61
4.4	Netzwerkprotokolle	62
4.5	TCP/IP Stack	64
4.6	XPort	65
4.6.1	Konfiguration	68
4.6.2	Implementierung	69
4.6.3	Speicherung der Webseiten	70
4.6.4	Web Server Grundlagen	70
4.6.5	Netzwerkumgebung und Serverzugriff	71
5	Massenspeicher zur Langzeitaufzeichnung	74
5.1	CSV Dateiformat	75
5.2	VNC1L	75
5.2.1	VDIP1 Modul	77
5.2.2	Schnittstellen	77
5.2.3	SPI des VNC1L	78
5.2.4	Firmware	80
5.2.5	Datenspeicherung	81
5.2.6	Beschränkungen	82
6	Gesamtschaltung	83
6.1	Benutzerschnittstellen Ein- und Ausgabe	83
6.1.1	Schaltung zur Entprellung der Tasten	83
6.1.2	LC Display	84
6.2	Spannungsversorgung und galvanische Trennung	84
6.2.1	LT3500	86
6.2.2	Digital Isolator	87
6.3	Schaltungslayout	89
6.3.1	PCB	89
6.3.2	Allgemeine Sicherheit, Kriech- und Luftstrecken	90
7	Software	94
7.1	Programm Mikrocontroller	94
7.1.1	Softwarestruktur	94
7.1.2	Beschreibung und genereller Ablauf	95
7.2	Java Applet	102
7.2.1	Socket	103
7.2.2	MVC	103
7.2.3	Programm und Klassen	104
7.3	Anwendungsspezifisches Datenübertragungsprotokoll	105
8	Schlußbemerkung und Ausblick	109
A	Schaltpläne, Layout und Aufbau	110

Abbildungsverzeichnis

1.1	Allgemeine Funktionsblöcke	15
1.2	Vereinfachtes Blockschaltbild	15
2.1	XC167 Blockschaltbild	17
2.2	Blockschaltbild RTC	20
3.1	Blockschaltbild ADE7758	24
3.2	Filterung des Rauschens	25
3.3	Eingangsstruktur Strom	26
3.4	Eingangsstruktur Spannung	26
3.5	Signalpfad Effektivwertberechnung	27
3.6	Filterung des quadrierten Signals	28
3.7	Wirkleistungssignal	29
3.8	Wirkleistungsberechnung	30
3.9	Vorzeichen von Blind- und Wirkleistung	30
3.10	Line Cycle Modus	31
3.11	Leistungs-dreieck	34
3.12	Power Faktor	35
3.13	SPI Bus	37
3.14	Darstellung der Betriebszustände	38
3.15	ADE7758 Kommunikationsstruktur	38
3.16	ADE7758 Schreibzyklus	39
3.17	ADE7758 Lesezyklus	39
3.18	Open Loop Hall Stromwandler	40
3.19	Schaltung eines Stromwandlers	41
3.20	Aliasing Effekt	42
3.21	Strom-,Spannungsrichtige Messung	52
3.22	Spannungsrichtige Kalibrierschaltung	53
3.23	Anschluß Sternschaltung	55
3.24	Anschluß Dreieckschaltung	56
3.25	Aronschaltung	56
4.1	OSI vs. Internetschichtmodell	58
4.2	Datenkapselung	59
4.3	Netzwerkhardware	60
4.4	Hardwareelemente	60
4.5	Komplettlösung Netzwerkhardware	61

4.6	Ethernetframe	62
4.7	Aufbau XPort	65
4.8	Blockschaltbild XPort	66
4.9	Softwareschichtmodell XPort	67
5.1	Blockschaltbild VNC1L	76
5.2	Lesezyklus SPI VNC1L	78
5.3	Schreibzyklus SPI VNC1L	79
5.4	Command Monitor VNC1L	80
6.1	Entprellschaltung	84
6.2	Versorgung	86
6.3	Schaltregler	87
6.4	Verzögerungen am SPI-Bus	88
6.5	Verwendung von zwei SPI-Schnittstellen	89
6.6	Erdimpedanz	90
6.7	Klassen von Isolierungen	91
6.8	Kiech- und Luftstrecke	92
7.1	Softwarestruktur Embedded System	95
7.2	Flussdiagramm des Programms	101
7.3	Ausführung Java Applet	102
7.4	Lebenszyklus des Java Applet	102
7.5	Kommunikationskanal Applet - XPort - Mikrocontroller	103
7.6	GUI Java Applet	105
7.7	Kommunikationsstruktur XPort - Java Applet	106
7.8	Datenframe	107
7.9	Ausschnitt Receive State Machine	108
A.1	Schaltplan XC167 Kern 1/4	111
A.2	Schaltplan XC167 Kern 2/4	112
A.3	Schaltplan XC167 Kern 3/4	113
A.4	Schaltplan XC167 Kern 4/4	114
A.5	Layout XC167 Kern	115
A.6	Bestückung Kern Top	116
A.7	Bestückung Kern Bottom	116
A.8	Schaltplan Messplatine 1/2	117
A.9	Schaltplan Messplatine 2/2	118
A.10	Layout Messplatine TOP	119
A.11	Layout Messplatine BOTTOM	119
A.12	Bestückung Messplatine TOP	120
A.13	Bestückung Messplatine BOTTOM	120
A.14	Schaltplan Traegerplatine 1/2	121
A.15	Schaltplan Traegerplatine 2/2	122
A.16	Layout Traegerplatine TOP	123
A.17	Layout Traegerplatine BOTTOM	123
A.18	Bestückung Traegerplatine TOP	124

A.19 Bestückung Trägerplatine BOTTOM	124
A.20 Schaltplan VDIP1	125
A.25 Luftstrecke EN 60950	128
A.26 Kriechstrecke EN 60950	129

Tabellenverzeichnis

2.1	Features und Vergleich XC167 mit C167.	19
3.1	Betriebszustände SPI Bus	37
3.2	Vergleich Stromsensoren	42
4.1	Tabelle Internet-Schichten, Funktionen, Protokolle	59
4.2	Features XPort	68
5.1	Konfiguration Schnittstelle VDIP1	77

Kapitel 1

Einleitung

1.1 Motivation

Noch nie war das Energiebewusstsein und die damit verbundene effiziente Nutzung größer und wichtiger als heute, da einerseits der Verbrauch nicht nur durch die industrielle Aufholjagd der östlichen Staaten und Asiens stärker als die bereitgestellte Versorgungskapazität ansteigt, sondern auch der Mensch nicht zuletzt im Zuge der Umweltdebatte erkannt hat, dass aufgrund beschränkter Ressourcen eine nachhaltige Energiepolitik notwendig ist. Potenzielle Engpässe führen zur Regulierung mit dem Ziel des Energiesparens und der Steigerung der Energieeffizienz. Zu beobachten ist diese Entwicklung auch daran, dass Maßnahmen im Sinne von Gesetzen und Direktiven durch die Politik beschlossen werden, die dem Erreichen dieses Zieles dienen. Die Europäische Union hat 2009 als Teil der Ökodesign-Richtlinie, zur Erhöhung der Effizienz und Entlastung der Umwelt, ob des schlechten Wirkungsgrades bereits mit der stufenweisen Abschaffung der klassischen Glühbirne begonnen.

Die Gesamtkosten eines Betriebes für elektrischen Strom entstehen aus dem Arbeitspreis für die konsumierte elektrische Arbeit (kWh) und den Leistungspreis für die Maximalleistung (kW). Der Leistungspreis und folglich die Gesamtkosten lassen sich minimieren, wenn Leistungsspitzen vermieden oder gesenkt werden. Diese treten durch zufällig aufeinandertreffende Stromspitzen auf, was durch ein intelligentes Lastmanagement anhand permanenter Messungen des Stromverbrauches erfasst wird. Durch entsprechendes Gegensteuern, wie durch Wegschalten von Verbrauchern, kann das Lastverhalten optimiert und Einsparungspotenzial aufgezeigt werden. Im Zuge dieser Arbeit soll ein modernes und flexibel konfigurierbares Energiemessgerät entstehen, das sich am Vorbild der heute verfügbaren Smart Meter orientiert.

Grundprinzip eines Energiezählers ist die ständige Erfassung und Darstellung gelieferter und/oder verbrauchter elektrischer Energie. Hauptsächlich werden heutzutage gerade z. B. im Heimbereich überwiegend noch Ferraris Stromzähler eingesetzt, die quasi die letzten elektromagnetischen Überbleibsel des Victorianischen Zeitalters sind. Heutzutage ist das Netz, nicht zuletzt wegen elektrischer Maschinen, die rund die Hälfte der verfügbaren elektrischen Energie konsumieren, großen Blindelementen und nichtlinearen Lasten ausgesetzt, deren Energieaufnahme von solchen elektromechanischen Messgeräten nicht hinreichend gut gemessen werden kann. Deshalb kommt es vermehrt zum Einsatz von elektronischen

Energiezählern, die die aufgenommenen Daten auch noch beliebig verarbeiten können.

1.2 Aufgabenstellung

Angeregt von der Firma Knapp ist das Ziel dieser Diplomarbeit die Entwicklung eines Smart Energy Meters zur Analyse der wichtigsten Netzparameter unter Berücksichtigung folgender Kriterien hinsichtlich der Spezifikation.

➔ Dreiphasennetz

Kompatibilität zu Stern- und Dreiecksschaltung

➔ Spannungsmessung

Effektivwertermittlung von Wechselspannung bis 400V

➔ Strommessung

Effektivwertermittlung von Wechselstrom bis 5A

➔ Energiemessung

Aufzeichnung von akkumulierter Wirk-, Schein-, und Blindenergie

➔ Leistungsmessung

Ermittlung der entsprechenden Leistungswerte der einzelnen Phasen

➔ Power Faktor

Ermittlung des Verhältnisses von Wirk-, zu Scheinleistung

➔ Frequenzmessung

Erfassung der momentanen Netzfrequenz

➔ RS232

Externe Schnittstelle für Programmier- und Servicetätigkeiten

➔ LC Display

Anzeige von Messwerten und Programmmenü

➔ USB Schnittstelle

Messdatenaufzeichnung auf externen USB Flash Speicher

➔ Embedded Web Server

Messdatenübertragung und Steuerung über Netzwerk und Internet

➔ Betriebsspannung

Versorgung mit 12 und 24 Volt

➔ Gehäuse

Kompakte Gesamtbauweise und Eingliederung in ein Gehäuse

1.3 Entwurf und Umsetzung

Die gesamte Applikation kann in einzelne Aufgabengebiete bezüglich der Hardware und der damit im Zusammenhang stehenden Software aufgeteilt werden. Deren Kombination soll die gewünschte Funktionalität gewährleisten. Unter Berücksichtigung der Schnittstellen sind die einzelnen Funktionsblöcke weitgehend unabhängig voneinander zu entwerfen.

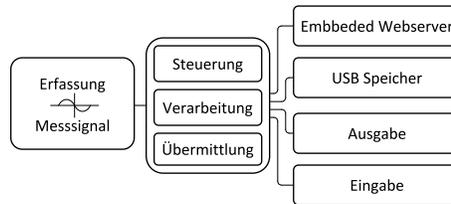


Abbildung 1.1: Allgemeine Funktionsblöcke des Systems.

Die zur Ermittlung der gewünschten Messgrößen nötigen Signale werden erfasst und von einer zentralen Einheit verarbeitet und weitergeleitet. Neben der obligatorischen Ausgabe der erfassten Daten über ein Display, sowie einer Eingabemöglichkeit über Tasten, soll weiters die Möglichkeit bestehen, die Daten in einer Datei auf einem USB Speichermedium unter Berücksichtigung des Dateisystems abzulegen. Zusätzlich sollen Daten auch über ein Netzwerk und das Internet übertragen werden. Insofern bietet sich neben der Möglichkeit das Messgerät aus der Ferne zu steuern, auch die Gelegenheit durch periodische Übertragung der aktuellen Messwerte, diese grafisch, als Funktion der Zeit darzustellen.

Zunächst sind also grundsätzliche, allgemeine Fragen zur effizienten und effektiven Erfüllung der Anforderungen zu untersuchen und beantworten. Die Funktionsblöcke zur Erfüllung der einzelnen Teilaufgaben werden der Reihe nach, aber doch mit Bedacht auf das Gesamtsystem entworfen. Ebenso gliedert sich diese schriftliche Arbeit. Für den besseren Überblick wird hier vorgehend, unter Beachtung der Funktionsblöcke, bereits das entsprechende Blockschaltbild der Gesamtschaltung mit den dafür wesentlichen Komponenten erstellt (Abbildung 1.2).

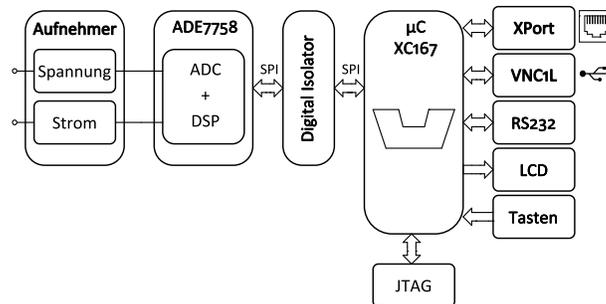


Abbildung 1.2: Vereinfachtes Blockschaltbild des Gesamtsystems.

Nachfolgend wird auf alle Komponenten, Aspekte und Methoden eingegangen, die direkt oder indirekt zur Erfüllung der Aufgabe relevant sind. Das Ende dieser Arbeit markiert ein Gerät, das der Spezifikation entspricht und in der gewünschten Art und Weise seine Aufgaben erfüllen kann.

Kapitel 2

Mikrocontroller

Erst stellt sich die Frage nach dem geeigneten Mikrocontroller bzw. DSP Prozessor, um als zentrale Schaltstelle, die einzelnen Komponenten die zur Erfüllung der gestellten Aufgabe nötig sind, zu koordinieren und die nötigen Funktionen auszuführen. Die analogen Signale der Messaufnehmer für Strom und Spannung müssen kontinuierlich in ausreichend schneller Zeit digital gewandelt, verarbeitet und ausgegeben werden. Die Frage der Signalverarbeitung ist mit der Frage nach dem geeigneten Prozessor verbunden. Zu bedenken ist natürlich, dass drüber hinaus noch weitere Aufgaben, Berechnungen und Kommunikationsabläufe durchgeführt werden sollen, womit aber das System noch nicht ausgelastet bzw. überfordert sein sollte, um sich nicht negativ auf die Gesamtausführungsgeschwindigkeit der einzelnen in Abhängigkeit zueinander liegenden Tasks auszuwirken.

2.1 XC167

Für die vorliegende Aufgabenstellung bietet sich vor allem eine kombinierte Lösung an, da hier sowohl die Vorzüge eines DSP's bei der Datensignalverarbeitung als auch die Vielseitigkeit und Flexibilität eines μC von Vorteil sind. Ein solcher 16-Bit-Controller ist der XC167 der Firma Infineon, der über eine MAC (Multiply and Accumulate) Unit verfügt, die ein Hauptbestandteil eines DSP's ist und viele Befehle wie z. B. eine Multiplikation in einem Taktzyklus ausführt. Die CPU ist ein, mit maximal 40 MHz getakteter C166S V2 RISC Core.

Anhand des Blockschaltbildes 2.1 lässt sich erkennen, dass der Mikrocontroller, über massig integrierte Peripherie, Ports und vor allem Schnittstellen verfügt. Ihnen kommt für die zu erstellende Applikation eine zentrale Bedeutung zu, da der Controller in seiner Prozessumgebung den zentralen Verarbeitungsknoten unter den zusätzlichen Bausteinen darstellt. So kann die Kommunikation mit den verbundenen Komponenten einfach und schnell durch die Hardware erledigt werden, was sonst in Software umgesetzt werden müsste. Über die JTAG-Schnittstelle kann sowohl das Programm auf den Flash Speicher übertragen, als auch auf Hardwareebene mittels, des OCDS (On Chip Debug System) zur Fehlererkennung debugged werden. Der Speicher ist nach der Von Neumann Architektur aufgebaut, die sich in einem gemeinsamen Adressraum für Programmcode und Daten sowie Register und I/O Ports äußert. Die Größe des integrierten Flash Speichers ist dabei vom jeweiligen Derivat abhängig. Diese unterscheiden sich sonst auch noch im Peripherieumfang. Das Exemplar

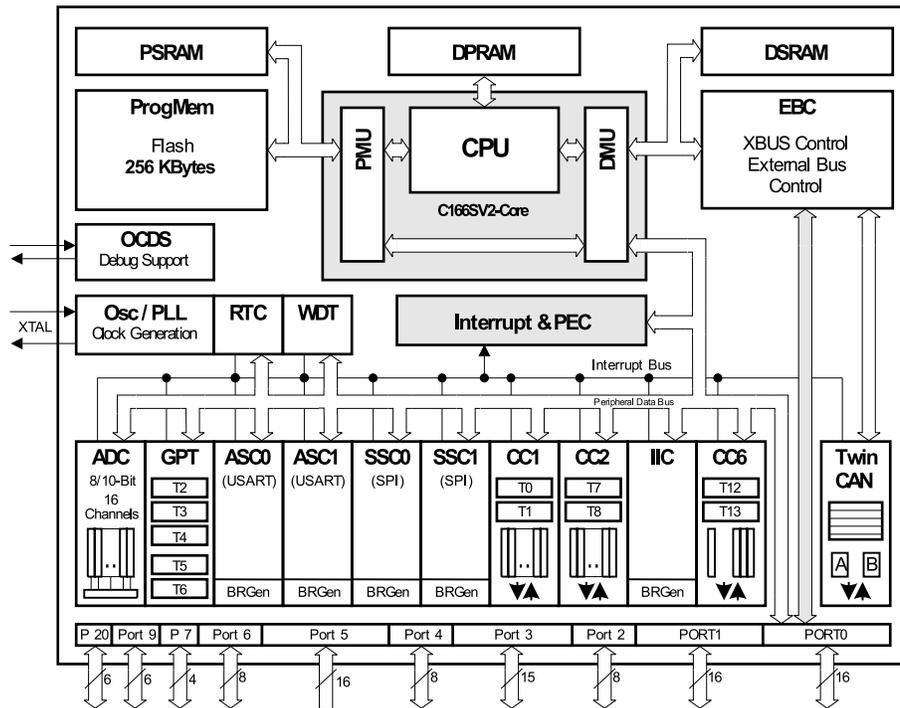


Abbildung 2.1: XC167 Blockschaltbild [V1.04a].

SAF-XC167CI-32F40F Grade A vereinigt, als umfangreichstes ausgestattetes Modell, alle möglichen Peripherieeinheiten und hat 256 KByte internen Flash Speicher und in Summe 12 KByte RAM on Chip. Grade A Devices haben im Gegensatz zum Standard Muster beim Zugriff auf den internen Flash einen, um einen Waitstate geringeren Wartezyklus. In typischen Applikationen reduziert sich die Performance durch den langsameren Zugriff bei einem zusätzlichen Waitstate zwischen 5-15%(vgl.[MB06]). Wie im Blockschaltbild ersichtlich, teilt sich der RAM-Speicher in drei Teile auf. Einerseits das Programm SRAM (PSRAM) zu 6 KByte, aus dem Programmcode ausgeführt werden kann, welcher Umstand für das Speichern ins Flash während der Programmausführung notwendig ist. Hinzu werden alle Routinen, die für diese Operation nötig sind, vom Programmspeicher im Flash ins RAM geladen und für die Dauer des Vorganges von dort aus exekutiert. Der restliche RAM-Speicher gliedert sich in 4 KByte Data SRAM (DSRAM) für die Speicherung volatiler Daten und 2 KByte Dual Port RAM (DPRAM) für den Systemstack, die Programmvariablen und die Registerbänke.(vgl.[V1.04a])

Am Institut für Elektronik wird vielfach für den Geräteentwurf mit Mikrocontrollern ein steckbarer Kern, der die nötigste Peripherie auf sich vereint, auf Basis des Controllers C167 eingesetzt. Ausgehend davon wurde im Zuge eines IT-Projekts (vgl.[Sup05]) und einer Diplomarbeit (vgl.[Kra07]) bereits ein Modul für das Nachfolgemodell XC167 entworfen und angepasst. Die im Rahmen dieser Arbeit vorgenommenen Modifikationen beschränken sich im Wesentlichen auf die Reduktion der vorhandenen Peripheriebausteine auf ein Minimum, sodass zur Integration in eine Applikation ein kleines Prozessormodul bereitsteht. Nachfolgend eine Tabelle mit einer Auflistung der Funktionalität und Besonderheiten des XC167

im Vergleich zum gebräuchlichen Vorgängermodell C167.

XC167 16-bit	C167 16-bit
5-stage execution pipeline and 2-stage instruction fetch pipeline	4-stage Pipeline
25 ns instruction execution time @40 MHz CPU clock	80 minimum instruction execution time @25 MHz CPU clock (60 ns @33 MHz)
25 ns multiplication (16 x16-bit) @40 MHz clock 1-Cycle Multiply-and-Accumulate (MAC)	400 ns multiplication (16 x16-bit) @ 25 MHz CPU clock (300 ns @ 33MHz)
525 ns (21 CPU clock cycles) fast background division (32-bit/ 16-bit) @ 40MHz clock	800 ns division (32-bit/ 16-bit) @25 MHz CPU clock (600 ns @33MHz)
12 MByte total linear address space for code and data	16 MByte total linear address space for code and data
2 Kbyte dual-port RAM for variables, register banks, system stack (no code)	3 Kbyte/ 2 Kbyte dual-port RAM for code and data (C167CS/C167CR)
4 Kbyte additional high-speed data SRAM (no code)	8 Kbyte/ 2 Kbyte on chip high-speed XRAM for code variables and user stack (C167CS/C167CR)
2 Kbyte high-speed program SRAM for code and data	
256 Kbyte advanced Program Flash Memory for instruction code and constant data	up to 32 Kbyte/ 128 Kbyte on-chip ROM (C167CS/C167CR)
73 interrupt nodes with separate interrupt vectors on 16 priority levels and on 8 group levels	up to 56 interrupt nodes with separate interrupt vectors on 16 priority levels and on 4 group levels
Fast external interrupts	Fast external interrupts
200 ns (8 cycle) typical interrupt latency in case of internal program and data execution	240 ns (180 ns)/ 400 ns (300 ns) typical/maximum interrupt latency in case of internal program execution @25 MHz (33 MHz)
8 channel Peripheral Event Controller (PEC)	8 channel Peripheral Event Controller (PEC)
16-Channel 10-bit A/D Converter with programmable conversion time (2.55 μ s minimum @10-bit), auto scan mode	24-Channel 10-bit A/D Converter with programmable conversion time (7.76 μ s minimum @10-bit), auto scan mode
Two Serial Interfaces (ASC0/1)	One Serial Interfaces (ASC0)
Two High Speed Synchronous Channel (SSC0/1)	One High Speed Synchronous Channel (SSC0)
Two Multifunctional General Purpose Timer Units (GPT1 and GPT2)	Two Multifunctional General Purpose Timer Units (GPT1 and GPT2)
Two 16-Channel Capture/Compare Units	Two 16-Channel Capture/Compare Units

IIC Bus module with 10-bit addressing and 400Kbit/s	
Real Time Clock (RTC) with dedicated oscillator and several interrupt possibilities	
Up to 103 IO Lines With Individual Bit Addressability	Up to 111 IO Lines With Individual Bit Addressability
Programmable driver characteristics	Programmable driver characteristics
Dual voltage power supply required: <ul style="list-style-type: none"> • 5V for I/O, ADC, and port structures (Vddp) • 2.5V for internal core, and XTAL (Vddi) 	
On Chip Debug Support (OCDS) via JTAG Interface	

Tabelle 2.1: Features und Vergleich XC167 mit C167[V2.03].

Als Entwicklungsumgebung wurde das am Institut für Elektronik eingesetzte Programm Keil Vision 3 verwendet, welches aus einem Assembler, C Compiler, Simulator und Debugger besteht. Für das hardwaremäßige Debuggen und Speichern ins Flash können diverse USB Wiggler mit USB/JTAG Interface direkt aus der Software heraus benutzt werden. Eine weitere Möglichkeit zum Speichern ins Flash bietet die freie Software Memtool von Infineon, mit der sich über die serielle Schnittstelle, mittels im Controller integrierten Bootstrap Loader, Programm-Code und Daten direkt ins interne Flash Memory des Mikrocontrollers schreiben lassen. Das dafür nötige HEX-File generiert Keil beim Kompilieren. Um den ASC Boot Loader zu aktivieren werden durch DIP-Schalter am Mikrocontrollerkern (switch 6 und 7 on) die Leitungen ALE und \overline{RD} über Pull-down Widerstände auf Low gezogen und so die Start-up Konfiguration nach einem Reset festgelegt. Ansonst liegen die Leitungen über interne Pull-up Widerstände an High Pegel. Als zusätzliche Alternative bietet sich noch das Monitorprogramm MiniMon an, das via Bootstrap Loader gestartet wird und unter anderem auch für das Beschreiben des Flash Speichers zu verwenden ist.

Mit dem von Infineon bereitgestellten Programm DAVE ist es möglich das Startup File und die Initialisierung der Peripherie einfach über ein grafisches Interface zu programmieren und ein Project File, welches automatisch generierten Code entspricht, mit den getroffenen Einstellungen für Keil zu erstellen.

Das Layout des Prozessormoduls beschränkt sich auf die wesentlichen Peripheriebausteine und ist durch möglichst kleine Steckverbindungen in eine andere Umgebung platzsparend zu integrieren. Der 4 lagige Aufbau, mit einem reinen Versorgungs- und Masse-Layer im Inneren verhilft der Schaltung neben, für Störungsimpulse niederohmigen Verbindungen zwischen Versorgungs- und Masseleiter, zu besonderer Störsicherheit. Möglich wird dies, unter Berücksichtigung der pulsformigen Stromaufnahme einer digitalen Schaltung, durch eine entsprechende Dimensionierung der Bypass- bzw. Entkoppel-Kondensatoren. Diese bilden mit parasitären Induktivitäten, ihren eigenen und den der Durchkontaktierungen Schwingkreise, die bei der Resonanzfrequenz die geringste Impedanz aufweisen. Diese Resonanzfrequenz wird durch die Wahl der Kapazitätswerte gleich der einfachen und/oder

der doppelten Basisfrequenz des digitalen Systems gemacht, da bei diesen Frequenzen die größten Störungen auftreten.

2.1.1 RTC

In Hinblick auf die Langzeitaufzeichnung von Messdaten und um diese übersichtlich und aussagekräftig am Computer zu bearbeiten, sollte der Datensatz zum Zeitpunkt seiner Aufzeichnung mit einem Zeitstempel versehen werden. Dieser identifiziert ihn dann eindeutig unter der ganzen Reihe der aufgenommenen Daten und erleichtert seine spätere Zuordnung oder Interpretation.

Diese Zeitverwaltung erledigt eine Realtime Clock, die selbstständig und unabhängig vom Prozessortakt oder dem Zustand des Prozessors arbeitet. Der Mikrocontroller XC167 integriert bereits eine solche RTC, die aus einer Kette von Teilern und Timern besteht. Deren Takt kann wahlweise vom Prozessortakt abgeleitet werden, oder von einem zusätzlichen Quarzoszillator kommen, der speziell für diese Aufgabe bestimmt ist. Die Timer zählen alle aufwärts und werden mittels der Wahl der Reload-Werte beim gegebenen Takt auf die gewünschte Zählperiode und damit verbundene Zeitdauer angepasst. Die Zeiteinheit von einer Sekunde lässt sich bei einer Taktfrequenz des Quarzoszillators von 32.768kHz und einem Reload-Wert des 16-Bit Timers von 0x8000 genau einstellen. Mit Kenntnis der genauen Oszillatorfrequenz kann durch Software die Genauigkeit noch erhöht werden, indem man aus der Differenz zur idealen Taktfrequenz die Anzahl der Taktzyklen ausrechnet, bei der der Fehler zur idealen Frequenz ± 1 Taktzyklus ist. Dieser kann dann entsprechend korrigiert werden, entweder indem man zur gegebenen Zeit einen Clock-Puls zufügt oder unterdrückt. (vgl.[V1.04b])

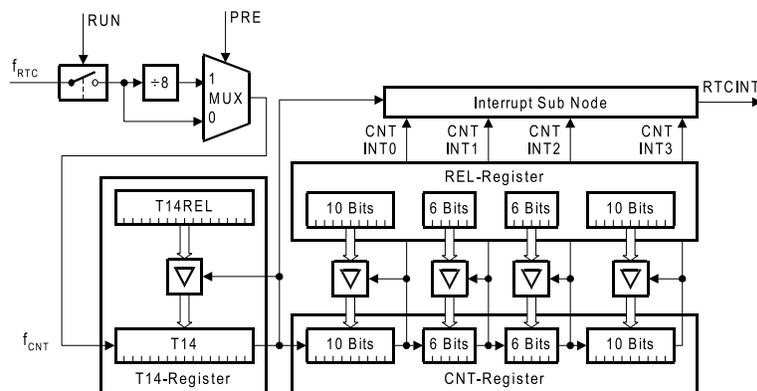


Abbildung 2.2: RTC Blockschaltbild [V1.04b].

Die Abbildung 2.2 zeigt ein Block Diagramm der RTC des XC167 mit den Zähl- und den Reload-Registern. T14 ist der Basis-Timer, der in dem durch den Reload-Wert bestimmten Zeitabstand bei Überlauf den nächsten Zähler anstößt, welcher beim Erreichen des bestimmten Zeitwertes wiederum das gleiche mit dem Nächsten macht. Es können zusätzlich auch Interrupts bei einem Überlauf ausgelöst werden. Da für die Zuordnung zu den Messwerten keine Zeitaufösung im Bereich von Millisekunden nötig ist, wurde als

Zeiteinheit eine Sekunde für den Basis-Timer T14 gewählt. Die folgenden Register haben eine Zeitbasis von einer Minute, einer Stunde und einem Tag. Da primär die Uhrzeit von Bedeutung ist, zu welchem Zeitpunkt die Daten aufgezeichnet wurden und nicht das momentane Datum, wurde auf Software, die Angaben über den Tag, das jeweilige Monat und das Jahr macht vorläufig verzichtet. In Bezug auf die Speicherung der Messwerte in einer Datei auf einem USB Speichermedium wäre dies aber durchaus sinnvoll, um die erstellte Datei mit dem Datum beziehungsweise mit der Uhrzeit des Erstellungszeitpunktes zu versehen. Nachdem das RTC-Control-Register und die Reload-Werte entsprechend der gewünschten Zeitbasis gesetzt sowie die RTC gestartet sind, hat man eine vom Zustand des Prozessors unabhängige fortlaufende Uhrzeit, die man immer zum Zeitpunkt der Datenaufzeichnung abfragen kann, um den gewünschten Zeitstempel zu erhalten.

Kapitel 3

Mess-Signalaufnahme und Verarbeitung

Zur Ermittlung der gewünschten Parameter bedarf es der Erfassung, der primären Messgrößen des Netzes, Spannung und Strom. Aus deren momentanen Messsignalen werden durch Signalverarbeitung bzw. Berechnungen weitere charakteristische Größen abgeleitet. Zielführend ist sicherlich der Einsatz eines Energy Metering IC's. Die am Markt verfügbaren Produkte unterscheiden sich in ihrem Funktionsumfang und bestimmten Einsatzzweck. Vom Mess-IC, der ausschließlich für Energiezähler und den Antrieb eines Schrittmotors über einen Pulsausgang konzipiert ist, über Produkte, deren digitale Signalverarbeitung über Register zugänglich ist, bis zu System-on-Chip Vertretern mit integrierten Mikrocontrollern, besteht eine große Bandbreite.

Ein Energy Metering IC kombiniert in jedem Fall genaue ADCs mit fix implementierter digitaler Signalverarbeitung in unterschiedlichem Umfang. Durch seine Konstruktion sind aufgenommene Abtastwerte von Spannung und Strom der einzelnen Phasen zeitlich kongruent, was für genaue weiterführende Berechnungen entscheidend ist. Folgende am Markt befindliche Produkte, verschiedener Hersteller, erscheinen prinzipiell für die gestellte Aufgabe gebräuchlich:

Microchip MCP 3909 Single Phase Mess-IC mit zwei 16 Bit Delta Sigma ADCs und einem Messfehler von 0.1% über einen dynamischen Bereich von 1000:1. Die momentanen Kurvenformen von Strom und Spannung, mit einer Bandbreite bis zu 14 kHz, und deren Produkt wird ermittelt und über SPI Schnittstelle ausgegeben.

ST STPM01 Single Phase Produkt zum Messen von Wirk-, Schein- und Blindenergie sowie den Effektivwerten von Strom und Spannung und den momentanen Kurvenformen bis zu einer Bandbreite von 400Hz mit einem Messfehler von 0.1%. Die Kommunikation erfolgt über die SPI-Schnittstelle.

Da bei der vorliegenden Anwendung Messwerte von 3Phasen aufgenommen werden sollen, ist der Nachteil von Single Phase Produkten offensichtlich. Es müssten drei Stück davon für die Messung an dreiphasigen Netztopologien eingesetzt werden.

CS5451A Dieser Baustein verfügt über sechs AD Wandler und digitale Filter, wobei die Abtastung der ADCs untereinander synchron erfolgt. Er bildet somit ein speziell

angepasstes Front End, doch hat die Signalverarbeitung in einem separaten DSP oder Mikrocontroller zu erfolgen, was wiederum zu großer Flexibilität aber auch Komplexität führt.

Analog Device ADE7758 Misst Schein-, Wirk- und Blindenergie, Effektivwerte von Spannung und Strom, momentane Kurvenformen und Frequenz, dreiphasiger Netztopologien. Interface ist ein SPI-Bus mit einer zusätzlichen Interruptleitung. Der Messfehler bei Wirkenergie liegt bei 0.1% über einen dynamischen Bereich von 1000:1.

MAXQ3180 Ermittelt Schein-, Wirk- und Blindenergie sowie Leistung, Effektivwerte von Spannung und Strom als auch die Harmonischen dieser Signale an bis zu sieben differenziellen Eingängen und ist somit für dreiphasige Topologien geeignet. Als Schnittstelle fungiert dabei der SPI-Bus.

Chips der ADE71xx und ADE75xx Serie verfügen zusätzlich zu ihrem fix implementierten Signalverarbeitungsteil, über einen integrierten Mikrocontroller auf 8052 Basis, was einen Vorteil hinsichtlich einer Komplettlösung darstellt, falls nur Messaufgaben und deren Ausgabe am Display bzw. andere einfache Tasks zu bewältigen sind. Vorteilhaft wirkt sich auch die Tatsache aus, dass die Daten des Analogteils dem Mikrocontroller sofort zur Verfügung stehen und nicht erst über eine Schnittstelle von diesem ausgelesen werden müssen. Weiters existieren Mikrocontroller, die über entsprechend ausgelegte A/D-Umsetzer verfügen, wie Derivate des MSP430F471xx von Texas Instruments und so bevorzugt für Messaufgaben im Stromnetz eingesetzt werden.

3.1 ADE7758

Features:

- Kompatibel zu 3 Phasen/ 3 Leiter und 3 Phasen/ 4 Leiter Topologien
- Schein-, Wirk-, Blindenergie, Spannung RMS, Strom RMS
- Momentane Kurvenformdaten für ein auswählbares Signal
- SPI Interface
- Periodenmessung mit Nulldurchgangsdetektion
- Einstellbares Interruptsystem
- Energieberechnung durch Integration über Halbwellen
- Spitzenwertdetektion
- SAG Detektion
- Temperaturmessung
- 2 Pulsausgänge für Wirk- und Blind-/ Schein-Leistung mit einstellbarer Frequenz
- On-chip Kalibrierungsregister

Die Abbildung 3.1 zeigt das gesamte Blockdiagramm des ADE7758, die Eingangsstruktur, die Signalpfade und das serielle SPI-Interface. Die einzelnen Strukturen, die mathematischen Prozesse und Abläufe zur Berechnung der Messwerte, werden nachfolgend genauer behandelt, um Verständnis für das Prinzip der Messung zu bekommen. Anhand des Blockschalbilds sollen auch die Zusammenhänge der einzelnen Signalwege und der damit verbundenen Berechnungen deutlich werden, um den Überblick zu vervollständigen.

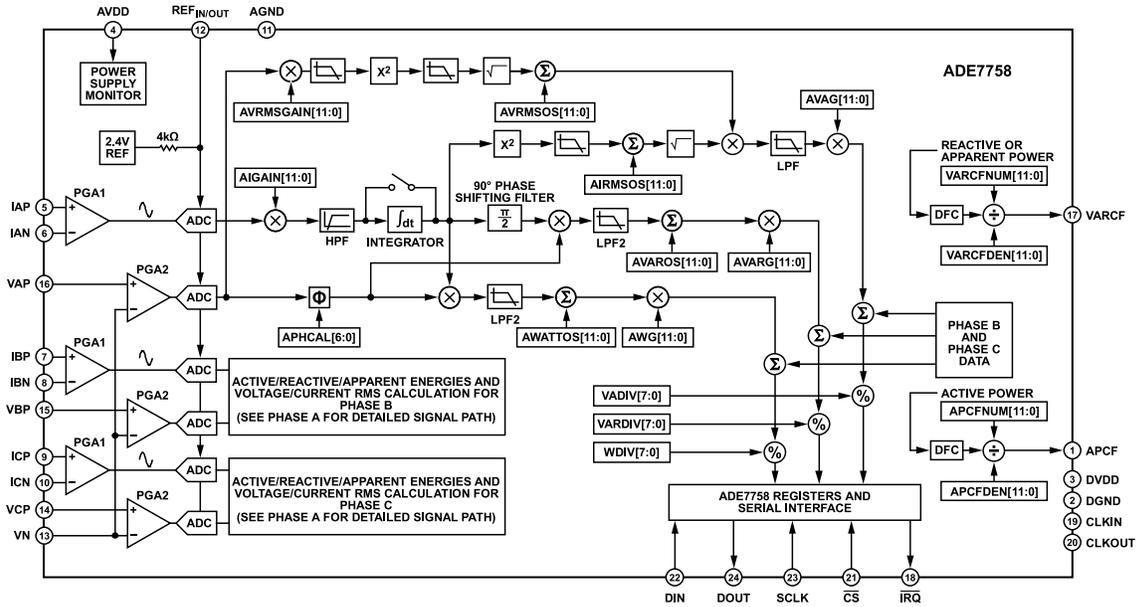


Abbildung 3.1: Blockschalbild ADE7758 [ADE06].

Um die Messaufnahme und die folgende Signalverarbeitung möglichst plakativ und nachvollziehbar zu zeigen, wird auf die entsprechenden Kapitel des Datenblattes etwas genauer eingegangen.

3.1.1 Analog-Eingänge

Der ADE7758 hat sieben analoge Eingänge für die Messsignalerfassung, drei für die Strom- und drei für die Spannungsmessung plus den Bezugsleiter. Die drei Stromkanäle bestehen aus drei fully differential Spannungseingängen: IAP/IAN, IBP/IBN, ICP/ICN. Das maximale differenzielle Eingangssignal an diesen Kanälen beträgt bei Full Scale $\pm 500\text{m}\hat{V}$ Spitzenwert und sollte, um den IC nicht zu beschädigen, nicht Werte jenseits $\pm 6\hat{V}$ annehmen. Durch geeignete Dimensionierung der Eingangsstruktur ist es möglich, auch die Gefahr einer Beschädigung des IC's durch zu hohe Eingangsgrößen zu minimieren. Das Eingangssignal kann durch einen PGA (Programmable Gain Amplifier) mit einem Multiplikationsfaktor in Stufen von 1, 2 und 4 verstärkt und für die anschließenden AD Wandler angepasst werden, um den Full Scale von $\pm 0.5\hat{V}$ für ein bestmögliches Messergebnis auszunutzen. Darüber hinaus bieten die Stromkanäle noch die Möglichkeit, diesen Full Scale auf $\pm 0.5\hat{V}$, $\pm 0.25\hat{V}$ oder $\pm 0.125\hat{V}$ einzustellen, was durch das Anpassen der AD Wandler Referenz erreicht wird. Die Spannungskanäle haben drei single-ended Spannungseingänge: VAP, VBP und VCP und den Referenzleiter VN. Diese Kanäle haben in Bezug auf VN

ebenfalls einen maximalen Eingangsspannungsbereich von $\pm 500\text{m}\hat{V}$ und bieten auch die Möglichkeit der Signalanpassung durch einen PGA. Die A/D-Konverter bestehen aus einem A/D-Wandler, der nach dem Sigma-Delta Prinzip arbeitet und einem digitalen Tiefpassfilter.

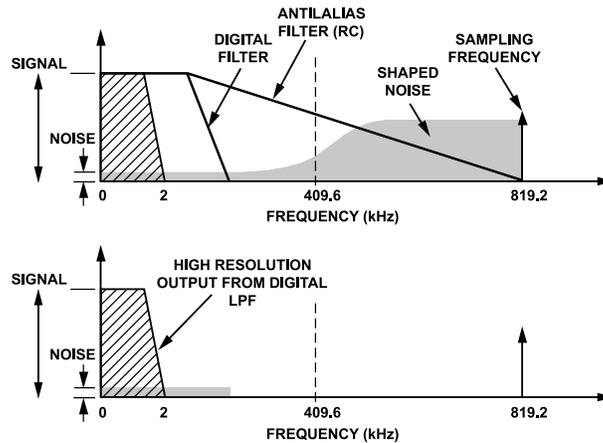


Abbildung 3.2: Filterung des Rauschens [ADE07].

Durch Überabtastung des Signals wird das Rauschen über eine größere Bandbreite gespreizt mit dem Effekt, dass es innerhalb der Bandbreite, die von Interesse für die Messung ist, geringer ausfällt. Der Integrator des Sigma-Delta Umsetzers, welcher Hochpasseigenschaften für das Quantifizierungsrauschen besitzt, formt es so, dass die Mehrheit davon bei höheren Frequenzen liegt und vom nachfolgenden Tiefpass entfernt wird (vgl. Abb. 3.2). So wird das Signal-Rausch-Abstandsverhältnis erhöht und die Auflösung und Genauigkeit verbessert.

Die Abbildung 3.3 zeigt einen Stromkanal Pfad mit dem PGA, ADC und einem Ausschnitt des Signalverarbeitungsteiles. Der ADC gibt den maximalen digitalen Ausgabewert bei einem analogen Eingangswert von $\pm 0,5\hat{V}$ aus. Die Kurvenform eines bestimmten Signals (momentane Signalform von Spannung, Strom, Schein-, Wirk-, Blind-Leistung einer definierten Phase) wird mit einer maximalen Frequenz von 26kSPS abgetastet und die erhaltenen Werte dabei mit 24 Bit aufgelöst.

Das Fullscale Signal am Eingang des differentiellen Stromkanals entspricht einem digital gewandelten Signal im Wertebereich von 0xD7AE14 (-2,642,412) und 0x2851EC (+2,642,412). Durch die Multiplikation mit einem Verstärkungs-Faktor kann dieses nun in einem Prozentbereich vom $\pm 50\%$ verändert werden. Zu beachten ist, dass durch diese Anpassung alle weiteren Berechnungen im Signalpfad, die auf diesem gewandelten Stromsignal beruhen, wie die von Energie und Effektivwert des Stromes, um eben diesen Faktor beeinflusst werden.

Der Hochpassfilter HPF zum Entfernen eines Gleichanteils des Signals kann deaktiviert werden, um auch Gleichströme zu messen. Der digitale Integrator dient zur Signal Rekonstruktion, wenn als Strommessaufnehmer ein di/dt Sensor verwendet wird, und ist andernfalls deaktiviert.

Die Eingangsstruktur des Spannungspfades, Abbildung 3.4, ist mit PGA und ADC ähnlich aufgebaut wie die des Strompfades. Es ist kein Hochpassfilter implementiert, da

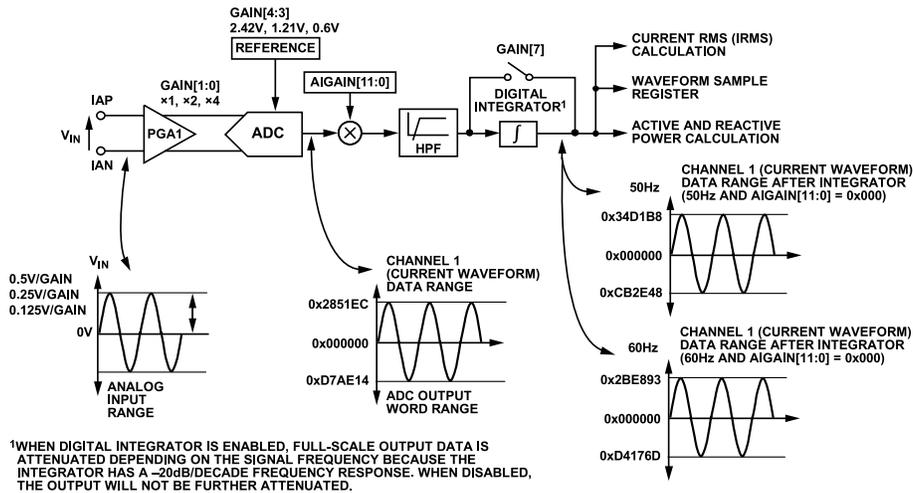


Abbildung 3.3: Eingangsstruktur Strom [ADE06].

der HPF im Stromzweig ausreichen sollte, um Fehler in der Leistungsberechnung, die auf ADC Offset zurückzuführen sind, zu eliminieren. Durch das Entfernen eines Offset an wenigstens einem Zweig entsteht bei der Multiplikation zur Leistungsberechnung keine Gleichkomponente. Der mit der Kreisfrequenz ω schwingende Fehleranteil wird dabei durch einen nachfolgenden Tiefpass weggefiltert. Ein ADC-Offset in den Spannungsweigen ruft jedoch Fehler in der Effektivwertberechnung hervor und beeinflusst im weiteren Verlauf auch die Genauigkeit der Messung der Scheinenergie. Für Wirk- und Blindenergiemessungen wird der Ausgang des ADC's über ein Phasenkorrekturglied zu den Multiplikatoren geführt. Dieses gleicht Phasenverschiebungen aus, die auf Stromwandler oder Anpassungsnetzwerke zurückzuführen sind, die einen unterschiedlichen Phasengang aufweisen. Das ist notwendig, um zeitlich kongruente Signale von Spannung und Strom zu erhalten und daraus genaue Leistungs- bzw. Energiewerte zu bekommen. Der Tiefpassfilter LPF1 hat eine Grenzfrequenz von 260Hz und schwächt das Ausgangssignal des ADC's frequenzabhängig leicht ab.

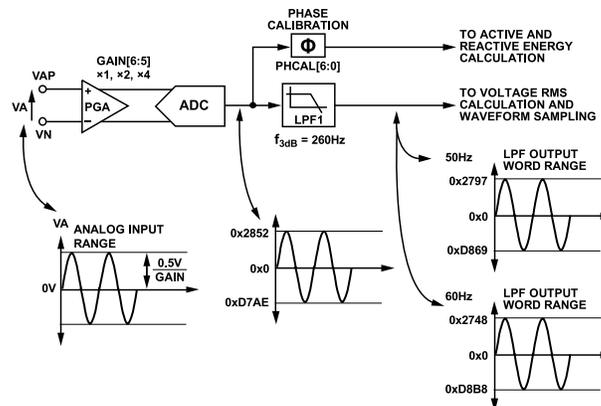


Abbildung 3.4: Eingangsstruktur Spannung [ADE06].

Diese Betrachtung des Strom- und Spannungskanals bildet die Grundlage für das Funk-

tionsprinzip aller weiteren Signalwege und zur Berechnung der gewünschten Größen wie Effektivwerte von Strom und Spannung sowie der Energie.

3.1.2 Effektivwertberechnung

Der Effektivwert RMS einer Wechselgröße entspricht jener Gleichgröße, die benötigt wird, um die gleiche Leistung an einer ohmschen Last umzusetzen.

Mathematisch ist der Effektivwert eines zeitkontinuierlichen Signals

$$FRMS = \sqrt{\frac{1}{T} \int_0^T f^2(t) dt} \tag{3.1}$$

und für ein zeitdiskretes Signal, wie es in einem abtastenden System vorliegt

$$FRMS = \sqrt{\frac{1}{N} \sum_{n=1}^N f^2[n]}. \tag{3.2}$$

$$\tag{3.3}$$

Die Methode, mit der im ADE7758 der Effektivwert für Strom und Spannung berechnet wird, ist das quadrierte Signal mit einem Tiefpass zu filtern und daraus dann die Wurzel zu ziehen.

Mathematisch gilt das folglich für ein Signal z. B. $v(t)$:

$$v(t) = \sqrt{2} \cdot VRMS \cdot \sin(\omega t) \tag{3.4}$$

$$v^2(t) = VRMS^2 - VRMS^2 \cdot \cos(2\omega t) \tag{3.5}$$

Der mit doppelter Kreisfrequenz schwingende Anteil wird weggefiltert. Die Abbildung 3.5

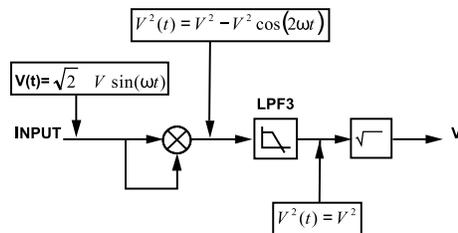


Abbildung 3.5: Signalpfad Effektivwertberechnung [ADE06].

zeigt einen Ausschnitt aus einem Signalpfad zur Darstellung der Effektivwertberechnung. Es wird nur der Spannungspfad gezeigt, da der Strompfad vom Prinzip gleich aufgebaut ist. Für den genauen Signalweg von Strom bzw. Spannung sei auf das vollständige Blockschaltbild Abb. 3.1 verwiesen, welches den gesamten Signalverlauf grafisch darstellt. Die so gewonnenen Effektivwerte können nun durch die Offsetkompensationsregister (xRMSOS) um ihren Offsetwert korrigiert werden (siehe Kalibrierung und Justierung). Anzumerken ist, dass durch die Offsetkalibrierung nicht erreicht werden kann, dass die VRMS-Register

einen Wert 0 beinhalten, wenn keine Spannung an den Eingängen des ADE7758 anliegt. Der Grund dafür ist Rauschen in der Effektivwertberechnung der Spannung, welches den nutzbaren Spannungsbereich zwischen Full Scale und 1/50 von Full Scale beschränkt. Durch die Justierung dieses Offset wird außerdem der Offset der Scheinleistung, die aus den Effektivwerten von Strom und Spannung berechnet wird, entfernt. Die Bandbreite der Messung von nicht sinusförmigen Signalen ist durch die internen Filter bei der Effektivwertbestimmung der Spannung sowie der Berechnung von Scheinleistung auf 260Hz beschränkt. Bei der Effektivwertbestimmung von Strom und der Berechnung von Wirkleistung beträgt die Bandbreite hingegen 14kHz.

In Abbildung 3.6 ist die Filterung im Frequenzbereich dargestellt. Da aber das Filter nicht ideal ist, bleibt am Effektivwertsignal ein kleiner Ripple. Weil dieser mit in die Effektivwertregister übernommen wird, sollten diese Register möglichst synchron zu einem auftretenden Nulldurchgang gelesen werden, da zu diesem Zeitpunkt die Welligkeit klein ist und der Wert in den Registern am exaktesten mit dem tatsächlichen RMS-Werten übereinstimmt. Auf dieses Detail wurde auch in der Softwareimplementierung Wert gelegt. Dabei werden detektierte Nulldurchgänge mittels zeitgleich dazu ausgelöstem Interrupt signalisiert.

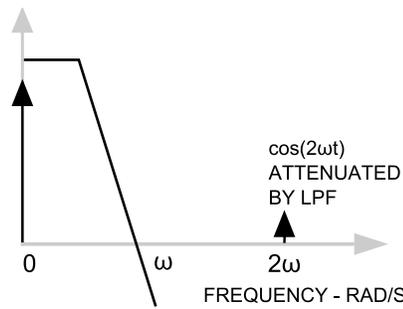


Abbildung 3.6: Filterung des quadrierten Signals

3.1.3 Wirkleistung

Das Produkt aus Spannungs- und Stromkurvenform, wie sie nach der Eingangsstruktur vorliegen, ergibt das momentane Leistungssignal, dessen Gleichgrößen-Komponente gleich der Wirkleistung ist.

$$v(t) = \sqrt{2} \cdot VRMS \cdot \sin(\omega t) \quad (3.6)$$

$$i(t) = \sqrt{2} \cdot IRMS \cdot \sin(\omega t) \quad (3.7)$$

$$p(t) = i(t) \cdot v(t) \quad (3.8)$$

$$p(t) = IRMS \cdot VRMS - IRMS \cdot VRMS \cdot \cos(2\omega t) \quad (3.9)$$

Die durchschnittliche Leistung über eine Anzahl n an Perioden ergibt sich aus

$$P = \frac{1}{nT} \int_0^{nT} p(t) dt = VRMS \cdot IRMS. \quad (3.10)$$

Die Abbildung 3.7 veranschaulicht den Signalverlauf graphisch. Die mit doppelter Kreisfre-

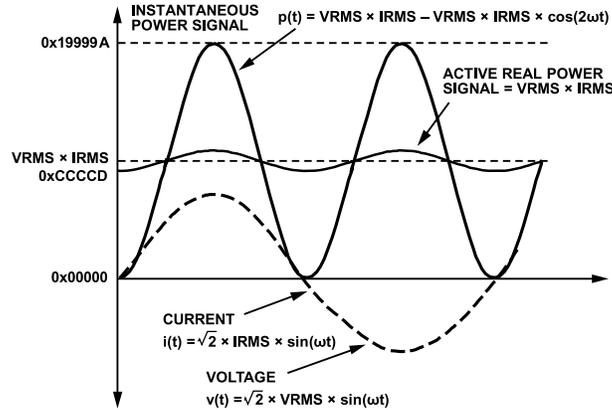


Abbildung 3.7: Wirkleistungssignal [ADE06].

quenz schwingende Komponente des momentanen Leistungssignals wird zur Wirkleistungsberechnung durch einen Tiefpass nahezu weggefiltert, um den Gleichanteil und damit die durchschnittliche Wirkleistung zu erhalten. Die Offsetkalibrierungsregister in diesem Signalpfad erlauben es, den Leistungswert um seinen Offset so zu kalibrieren, dass das Signal nahezu 0 ist, wenn keine Leistung konsumiert wird. Rauschen verursacht hier aber eine statistische Abweichung. Mithilfe der Gain Register xWG kann das Signal im Bereich von $\pm 50\%$ skaliert werden. Die $xDIV$ Register dienen zur Verlängerung der Zeit, bevor es zu einem Overflow der Energieregister kommt. Der sinusförmige Ripple, den das durchschnittliche Leistungssignal durch die nicht ideale Filterung enthält, wird durch die Integration, die zur Energieberechnung durchgeführt wird, eliminiert.

Energie ist definiert als Integral von Leistung über Zeit:

$$\text{Energie} = \int p(t)dt \quad (3.11)$$

Im diskreten System entspricht das Integral einer Summe:

$$\text{Energie} = \lim_{T \rightarrow 0} \left\{ \sum_{n=0}^{\infty} p(nT) \cdot T \right\} \quad (3.12)$$

Die Integration des Wirkleistungssignals erfolgt durch kontinuierliches Akkumulieren des durchschnittlichen Wirkleistungswertes pro Zeiteinheit in einem 41-Bit großen Register. Die oberen 16-Bit dieses Registers sind als Wirkleistungsregister vom Benutzer einsehbar. Die Abbildung 3.8 veranschaulicht den Signalpfad sowie die kontinuierliche Summation der durchschnittlichen Wirkleistung zur Wirkenergie.

Die Leistungsberechnung ist eine signed Operation. Bei einer Phasendifferenz von mehr als 90° wird die durchschnittliche Leistung negativ was bedeutet, dass Energie ins Netz zurückgespeist wird. Demzufolge ist auch das kontinuierliche Aufsummieren zur Energieberechnung eine signed Operation, sodass negative Werte vom Energieregisterinhalt subtrahiert werden. Wie in der Abbildung 3.9 ersichtlich, folgt für eine Phasenverschiebung von Strom und Spannung ein entsprechendes Vorzeichen von Wirk- und Blindleistung.

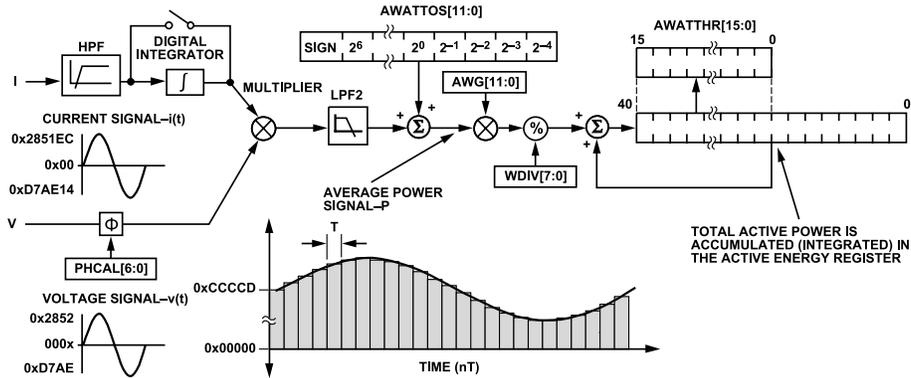


Abbildung 3.8: Wirkleistungsberechnung [ADE06].

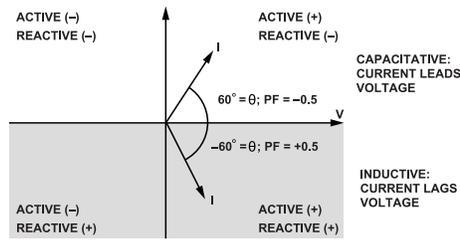


Abbildung 3.9: Vorzeichen von Blind- und Wirkleistung [Kap03].

Es muss darauf geachtet werden, dass die Energierregister gelesen werden, bevor ein Overflow oder Underflow auftritt. Nach dem Lesevorgang werden diese Register automatisch wieder zurückgesetzt. Die diskrete Periodendauer, mit der die Integration bzw. Addition in den Energierregistern erfolgt, beträgt $4\mu\text{s}$ ($4/\text{CLKIN}$). Mit einem Fullscale Signal an den Analogeingängen des ADE7758 unter stetigen Lastbedingungen und einem Inhalt der Gain-Register von Null, ergibt sich eine durchschnittliche Zeit bis zum Überlauf der signed Register von Wirk- und Scheinenergie von 0.52s und der unsigned Scheinenergieregister von 1.15s. Diese Zeit kann durch das Setzen der entsprechenden Divisionsregister um den gesetzten Wert verlängert werden.

Der ADE7758 verfügt über einen internen no-load-Threshold an jeder Phase, welcher die Energieakkumulation unterbricht, wenn die Wirkleistung unter 0.005% des Fullscale Input fällt.

Wenn der ADE7758 im Line Cycle Mode arbeitet, werden alle Energierregister gleichzeitig nach einer festzulegenden Anzahl an Nulldurchgängen der ausgewählten Phasen, durch die 41-Bit breiten Integrationsregister aktualisiert und ein Interrupt ausgelöst (vgl. Abb. 3.10). Die Periodendauer, mit welcher die Energierregister aktualisiert werden, ergibt sich durch folgende Gleichung:

$$\text{AccumTime} = \frac{\text{LINECYC}}{2 \cdot \text{Frequenz} \cdot \text{Phasen}} \quad (3.13)$$

LINECYC Anzahl der Halbwellen
Phasen Anzahl der selektierten Phasen

Der Wert des LINECYC Registers wird erst nach dem Setzen der Bits, die die Energie-

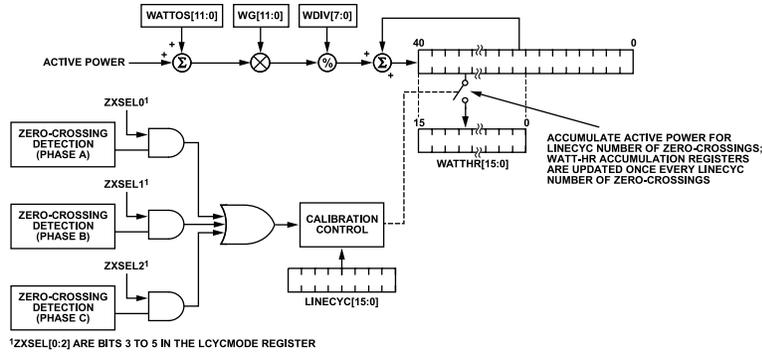


Abbildung 3.10: Line Cycle Modus [ADE06].

register in den Line Cycle Modus schalten, geladen, da dadurch der Nulldurchgangszähler des ADE7758 zurückgesetzt und so sichergestellt wird, dass das erste Akkumulationsergebnis korrekt ist.

Der Umstand, im Line Cycle Modus bei Kenntnis der Frequenz auch die Integrationszeit zu kennen ist nicht nur für die Kalibrierung der Energie hilfreich, sondern kann auch dazu verwendet werden, die Leistung und den Powerfaktor zu bestimmen, da dann sowohl Wirk- als auch Scheinleistung über die exakt gleiche Zeitspanne zu Energie aufsummiert werden.

Da der ADE7758 keine Register besitzt, in denen explizit durchschnittliche Leistungswerte gespeichert werden, müssen diese Werte im Mikrocontroller daher mit Kenntnis der genauen Integrationszeit aus den gewonnenen Energiewerten berechnet werden. Bei der Wahl der Anzahl von Halbwellen ist zu beachten, dass hier ein Kompromiss zwischen der Größe der durch längerer Integrationszeit resultierenden Werte, einem zufällig verteilten Fehler, durch den die Streuung um den tatsächlichen Wert bei längerer Integrationsdauer zunimmt, und der Zeit zwischen dem Auslesen der aktuellen Werte zur Ermittlung der durchschnittlichen Leistung, gefunden werden muss.

3.1.4 Blindleistung

Eine Last die Blindelemente beinhaltet, (Induktivitäten oder Kapazitäten) produziert eine Phasendifferenz zwischen der angelegten Spannung und dem resultierenden Strom. Die Blindleistung ist definiert als Produkt von Spannungs- und Stromkurve, wenn eines der beiden Signale um 90° phasenverschoben wird.

Aus diesem Grund beinhaltet der Signalpfad des Stromes, der zur Berechnung der Blindenergie führt, einen 90° Phasen Shift Filter (siehe Blockdiagramm Abb.: 3.1).

$$v(t) = \sqrt{2} \cdot VRMS \cdot \sin(\omega t - \theta) \quad (3.14)$$

$$i(t) = \sqrt{2} \cdot IRMS \cdot \sin(\omega t) \quad (3.15)$$

$$i'(t) = \sqrt{2} \cdot IRMS \cdot \sin(\omega t + \frac{\pi}{2}) \quad (3.16)$$

Das momentane Blindleistungssignal $q(t)$ wird durch die Multiplikation des Spannungssignals und des um 90° phasenverschobenen Stromes $i'(t)$ erzeugt.

$$q(t) = v(t) \cdot i'(t) \quad (3.17)$$

$$q(t) = V \cdot I \cos(-\theta - \frac{\pi}{2}) - V \cdot I \cos(2\omega t - \theta - \frac{\pi}{2}) \quad (3.18)$$

$$q(t) = V \cdot I \sin(\theta) + V \cdot I \cos(2\omega t - \theta) \quad (3.19)$$

Die durchschnittliche Blindleistung über eine Anzahl n an Perioden:

$$Q = \frac{1}{nT} \int_0^{nT} q(t) dt = VRMS \cdot IRMS \sin(\theta) \quad (3.20)$$

Die Gleichgrößen-Komponente des momentanen Blindleistungskurvensignals wird wie bei der Wirkleistung auch mit einem Tiefpassfilter LPF2 extrahiert, und zur Energieberechnung in einem 41-Bit großen Register aufaddiert, dessen oberen 16 Bit durch den Anwender gelesen werden können. Dabei wird auch der mit doppelter Frequenz schwingende Ripple, der durch den nicht idealen Tiefpass am durchschnittlichen Blindleistungssignal entsteht, eliminiert.

$$\text{Blindenergie} = \int q(t) dt \quad (3.21)$$

$$\text{Blindenergie} = \lim_{T \rightarrow 0} \left\{ \sum_{n=0}^{\infty} q(nT) \cdot T \right\} \quad (3.22)$$

Das durchschnittliche Blindleistungssignal kann vor der Akkumulation, durch das xVARG Register noch um $\pm 50\%$ skaliert und der Offset, durch Addition eines entsprechenden Wertes des Offsetregisters abgeglichen werden. Weiters besteht auch noch die Möglichkeit, durch einen Divisionsfaktor das Signal so anzupassen, dass die Zeit bis zum Überlauf des Blindenergieregisters um maximal das 255-fache (8bit Wert) verlängert werden kann.

3.1.5 Scheinleistung

Scheinleistung ist definiert als Produkt von Spannungseffektivwert und Stromeffektivwert (arithmetischer Ansatz)

$$S = VRMS \cdot IRMS, \quad (3.23)$$

oder als geometrische Summe von Wirkleistung und Blindleistung bzw. als die Wurzel der Summe von quadrierter Wirkleistung und quadrierter Blindleistung (vektorieller Ansatz)

$$S = \sqrt{P^2 + Q^2}. \quad (3.24)$$

Da im ADE7758 die Effektivwerte für Strom und Spannung bereits explizit vorliegen, ist zur Scheinleistungsberechnung der arithmetische Ansatz implementiert. Wie im Signalpfad ersichtlich ist, wird der Ausgang des Multiplikators mit einem Tiefpass gefiltert, um den

Ripple den die beiden Signale nach der Effektivwertberechnung aufweisen, bestmöglich zu eliminieren. Durch die xVAG Register kann dieses Signal wiederum im Bereich $\pm 50\%$ Prozent skaliert werden, bevor es zur Energieberechnung im 41 Bit breiten Register aufsummiert wird.

$$\text{Scheinenergie} = \int s(t) dt \quad (3.25)$$

$$\text{Scheinenergie} = \lim_{T \rightarrow 0} \left\{ \sum_{n=0}^{\infty} s(nT) \cdot T \right\} \quad (3.26)$$

Eine Offsetkalibrierung ist hier allerdings nicht mehr nötig, da in den Signalpfaden der Effektivwertberechnung diese bereits separat für Spannungs- und Stromeffektivwert erfolgt, und das Produkt dieser Werte keinen weiteren Offset erzeugt. Wie auch bei den anderen Energieregistern kann auch hier, durch einen wählbaren Divisor, die Zeit, bis das Scheinenergieregister überläuft, zeitlich verlängert werden.

3.1.6 Periodendauermessung/Frequenzmessung

Der ADE7758 ist in der Lage, mittels der erkannten Nulldurchgänge der einzelnen Phasen die Periodendauer zu messen. Dabei kann immer nur eine Phase ausgewählt werden, da nur ein Register zur Verfügung steht, in dem die Werte abgelegt werden können. Dieses Register wird alle vier Perioden der ausgewählten Phase aktualisiert, wobei auch ausge sucht werden kann, ob nun die Frequenz oder die Periode angezeigt wird. Wenn die Periode gemessen werden soll, beträgt die Auflösung des Registers $96/\text{CLKIN}$, was bei einem verwendeten Takt von 10 MHz $9.6\mu\text{s}/\text{LSB}$ entspricht. Bei Frequenzmessung entspricht ein LSB des Registers 0.0625Hz .

Da die Integrationszeit zur Ermittlung der Energie durch eine gewisse Anzahl an Halbwellen der Spannung bestimmt wird, ist die Erfassung der Netzfrequenz sowie die Kenntnis über die Anzahl an Phasen, bei denen die Zählung der Halbwellen erfolgt, zur Bestimmung der Zeit nötig.

3.1.7 Power Faktor

Der Power Faktor ist definiert als Verhältnis von Wirkleistung zu Scheinleistung und entspricht dem Betrag von $\cos(\varphi)$ bei sinusförmigem Strom- und Spannungsverlauf, wobei φ der Phasenwinkel zwischen Strom und Spannung bzw. der Winkel zwischen Wirkleistungs- und Scheinleistungsvektor ist. Dies wird in der Abbildung 3.11 veranschaulicht.

Der ADE7758 berechnet den Power Faktor nicht, also erfolgt die Berechnung im Mikrocontroller gewissermaßen nach der Wattmeter - Amperemeter - Voltmeter Methode.

$$\frac{P}{S} = |\cos \varphi| \quad (3.27)$$

Die Leistung steht ebenfalls nicht explizit durch ein Register zur Verfügung, aber sehr wohl Wirkenergie und Scheinenergie. Es muss aber sichergestellt werden, dass die Akkumulierungszeit dieser Energien zum Zeitpunkt des Auslesens der Register die Gleiche ist, damit die Formel auch für Wirk- und Scheinenergie gilt und sich somit der Faktor Zeit herauskürzt. Dies geschieht im Line Cycle Mode, in dem die Energieregister nach einer

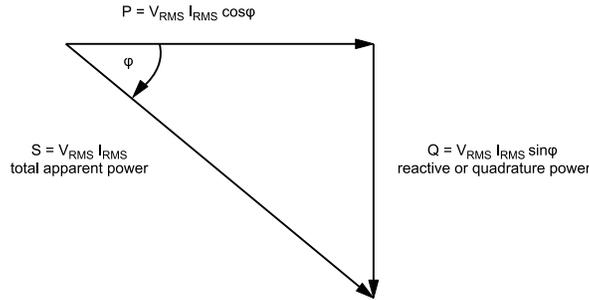


Abbildung 3.11: Leistungsdreieck [Wui03].

bestimmbaren Anzahl an Nulldurchgängen durch die oberen 16 Bit des Akkumulierungsregisters aktualisiert werden.

$$\frac{xWATTHR}{xVAHR} = |\cos \varphi| \tag{3.28}$$

Explizit sei hier noch auf den Unterschied zwischen Power Faktor und $\cos(\varphi)$ hingewiesen. Per Definition sind diese Größen nur identisch bei rein sinusförmigen Strom- und Spannungssignalen. Wenn Netzgeräte oder generelle Geräte zur Spannungsversorgung aber einen puls förmigen Strom liefern, beeinflusst nicht nur die Phasenverschiebung φ , sondern auch die enthaltenen Oberwellen des puls förmigen Signals die verfügbare Wirkleistung des Netzes. Der Power Faktor sagt also aus, wie sehr die Leistungsfähigkeit des Netzes sowohl durch die Phasenverschiebung, als auch durch die Oberwellen des Eingangssignals beeinflusst wird.

Die Fouriertransformierte eines periodischen nicht sinusförmigen Stromsignals ist:

$$IRMS_{total} = \sqrt{I_0^2 + I_{1RMS}^2 + I_{2RMS}^2 \dots + I_{nRMS}^2} \tag{3.29}$$

- I_0 ...Gleichanteil
- I_{1RMS} ...Grundwelle
- I_{nRMS} ...Oberwellen

Der Gleichanteil eines Wechselsignals ist null und die Grundwelle hat eine in-phase Komponente $I_{1RMS P}$ und eine Quadraturkomponente $I_{2RMS Q}$ dass geschrieben werden kann:

$$IRMS_{total} = \sqrt{I_{1RMS P}^2 + I_{2RMS Q}^2 + \sum_{n=2}^{\infty} I_{nRMS}^2} \tag{3.30}$$

Somit folgt für Wirkleistung:

$$P = V_{RMS} \cdot I_{1RMS P} \tag{3.31}$$

Mit φ_1 als Winkel zwischen Eingangsspannung und In-Phase Komponente der Grundwelle des Stroms ergibt sich:

$$I_{1RMSP} = I_{1RMS} \cdot \cos(\varphi_1) \tag{3.32}$$

$$P = V_{RMS} \cdot I_{1RMS} \cdot \cos(\varphi_1) \tag{3.33}$$

Aus der Scheinleistung

$$S = V_{RMS} \cdot IRMS_{total} \tag{3.34}$$

und der Definition für den Powerfaktor PF

$$PF = \frac{P}{S} = \frac{I_{1RMS} \cdot \cos(\varphi_1)}{IRMS_{total}} \tag{3.35}$$

folgt unter Einführung des Verzerrungswinkels θ , der endgültige Ausdruck für den Powerfaktor:

$$PF = \cos(\theta) \cdot \cos(\varphi_1) \tag{3.36}$$

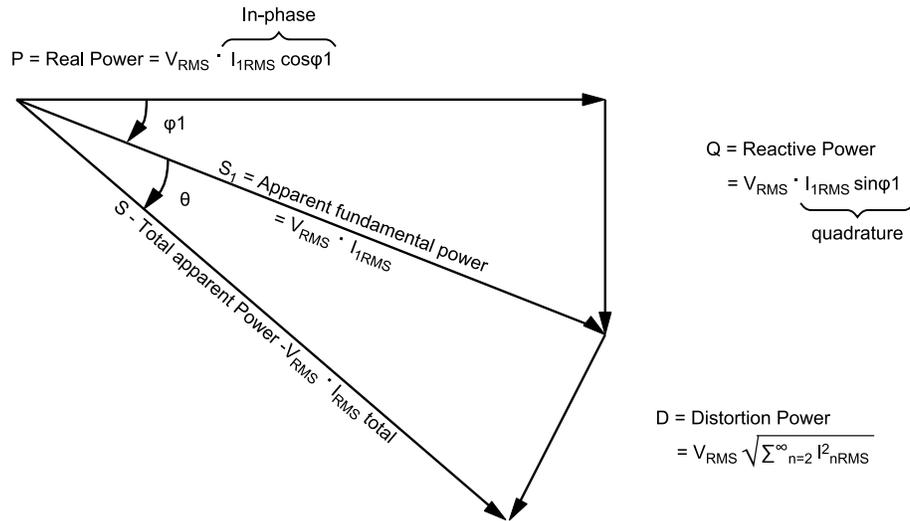


Abbildung 3.12: Power Faktor [Wui03].

Da der Powerfaktor, durch die Wirk- und Scheinleistung berechnet wird und die anliegende Stromsignalform innerhalb der aufgenommenen Bandbreite Teil der Leistungsbe-
 rechnung ist, ist im Ergebnis die Verzerrung aufgrund nicht sinusförmiger Stromsignale
 berücksichtigt.

3.1.8 Takt

CLKIN ist der Eingang des Taktsignals und der Haupttakt für die A/D-Umsetzer und
 Signal verarbeitenden Blöcke. Eine Eingangsfrequenz von 5Mhz Minimum bis 15Mhz Ma-
 ximum ist prinzipiell möglich, doch alle Spezifikationen und Performancewerte gelten für

eine Frequenz von 10Mhz. Eine Abweichung verändert die, nach dem Datenblatt gültigen Parameter und die Auflösung der Register. Eine externe Taktquelle kann an CLKIN angeschlossen werden oder ein Quarz zwischen CLKIN und CLKOUT, der dann die Quelle der Taktsignale darstellt. CLKOUT kann eine externe CMOS Last treiben, wenn entweder ein externer Taktgeber an CLKIN oder ein Quarz verwendet wird. In der Schaltung kommt ein 10Mhz Quarz zur Verwendung, der an beiden Enden über einen 22pF Lastkondensator an die digitalen Masse angeschlossen ist. Die Oszillatorschaltung schwingt mit der Frequenz, die durch die Resonanz von Quarz und Kondensatoren bestimmt wird.

3.1.9 Interrupts

Die Interrupts des ADE7758 werden über ein 24 Bit Interrupt-Status-Register und ein 24 Bit Interrupt-Mask-Register gesteuert. Im Interruptfall wird das entsprechende Bit im Interrupt-Status-Register gesetzt und überprüft, ob auch das äquivalente Bit im Interrupt-Mask-Register vom Benutzer gesetzt wurde. Ist dies der Fall, geht der Interruptausgang \overline{IRQ} aktiv low. Die Flags des Interrupt-Status-Registers werden aber unabhängig von den Bits des Mask-Registers gesetzt. Tritt ein Interrupt auf, muss der Mikrocontroller das Interrupt-Status-Register auslesen. Dies geschieht über das Reset-Interrupt-Status-Register, das ein eigenes Register aber mit demselben Inhalt des Interrupt-Status-Register ist, mit dem Unterschied, dass nach dem Lesen dieses Registers sowohl alle Flags des Reset-als auch des Interrupt-Status-Registers rückgesetzt werden. Der Interruptausgang wird bereits wieder high, nachdem der ADE7758 das Lesekommando (siehe SPI Schnittstelle) durch den ADE7758 empfängt. Er ist und bleibt dann garantiert solange high, bis das letzte Byte aus dem Reset-Interrupt-Status-Register ausgelesen ist, wobei auch in diesem Zeitraum kein Interrupt verloren geht, da der Ausgang nach dem Lesen des letzten Bytes sofort wieder auf low geht, falls inzwischen ein Ereignis eingetreten ist.

Der \overline{IRQ} Ausgang wird mit einem auf die fallende Flanke getriggerten Interrupteingang des Mikrocontrollers verbunden. Der XC167 besitzt einzelne Port-Leitungen, die im Fast-Modus abgetastet werden können und somit wesentlich kürzere Reaktionszeiten liefern, was bei zeitrelevanten Anwendungen von Vorteil ist. Sofort nach dem Detektieren der negativen Flanke startet der Mikrocontroller die Ausführung der Interrupt Service Routine (ISR), in der als Erstes weitere Interrupts ihren Priorität entsprechend gesperrt sowie das interne Interruptflag des μC wieder zurückgesetzt werden. Beim XC167 ist dies nicht explizit notwendig, da dies automatisch beim Einstieg in die ISR geschieht. Durch das Lesen des Reset-Interrupt-Status-Registers wird der \overline{IRQ} Ausgang high und der Mikrocontroller kann aus den gelesenen 24 Bit im Vergleich mit dem Interrupt-Mask-Register das auslösende Ereignis evaluieren. Tritt während der Ausführung der ISR wieder ein Interrupt ein und \overline{IRQ} geht auf low, wird dies durch das Setzen des Interruptflags aufgezeichnet und der μC kehrt gleich wieder in die ISR zurück, sobald die aktuelle ISR abgearbeitet ist und die Interrupts nicht weiter gesperrt sind. Der \overline{IRQ} Ausgang des ADE7758 ist ein open-drain Ausgang, was einen Pull-up Widerstand ($10k\Omega$) erfordert.

3.1.10 SPI Schnittstelle

Das Serial-Peripheral-Interface ist ein synchrones serielles Busprotokoll, welches von Motorola entwickelt wurde. Es benötigt üblicherweise zwei Steuerleitungen (CS und SCLK)

und zwei Datenleitungen, die meist mit MOSI Master-Out-Slave-In und MISO Master-In-Slave-Out bzw. SDO Serial-Data-Out und SDI Serial-Data-In bezeichnet werden. Über die CS Leitung, die gewöhnlich active-low ist, wird der anzusprechende Baustein selektiert. Im unselektierten Zustand ist die SDO Leitung tristate, was den Anschluss mehrerer Slaves erlaubt. SCLK als Clockleitung synchronisiert die Kommunikation und wird immer vom Master des Systems generiert.

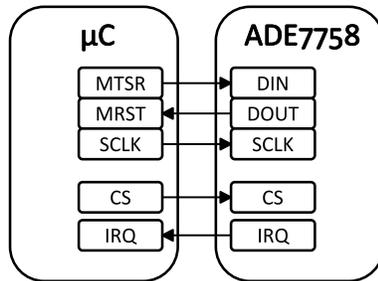


Abbildung 3.13: Schematische Darstellung des SPI Bus

Es ist kein fixes Übertragungsprotokoll vorgeschrieben, vier Betriebsarten sind aber standardisiert. Eine Übersicht der Betriebszustände bietet Tabelle 3.1 und Abbildung 3.14.

Clk Phase	Clk Polarität	
0	0	Idle Clock ist low, Daten werden bei high-low Flanke übernommen, und bei low-high Flanke geshiftet
0	1	Idle Clock ist low, Daten werden bei low-high Flanke übernommen, und bei high-low Flanke geshiftet
1	0	Idle Clock ist high, Daten werden bei low-high Flanke übernommen, und bei high-low Flanke geshiftet
1	1	Idle Clock ist high, Daten werden bei high-low Flanke übernommen, und bei low-high Flanke geshiftet

Tabelle 3.1: Betriebszustände SPI Bus

Der ADE7758 seinerseits verlangt eine Clock Phase und eine Clock Polarität von Null. Daten in Form von Bits werden am Ausgang DOUT mit der steigenden Flanke des Clock-signalen hinausgeschoben und am Eingang DIN, anhand der anliegenden Signalpegel mit der fallenden Flanke übernommen. Der Zustand von DOUT ist normalerweise hochohmig, solange keine Kommunikation am Bus stattfindet. Das MSB wird zuerst übertragen. Die maximale Baudrate beträgt 10Mbaud, wie der Timing Charakteristik zu entnehmen ist. Die Daten werden Byte-weise übertragen, auch wenn die meisten Register ein ganzzahliges Vielfaches größer sind. Innerhalb eines Schreib-/Lesezyklus (Zeit zwischen fallender und steigender Flanke von CS) werden diese 16 und 24 Bit langen Daten durch multiples Schreiben/Lesen von Bytes geschrieben/gelesen und vom Mikrocontroller entsprechend aufbereitet. Bei Registern, die kein ganzzahliges Vielfaches breiter oder aber auch kleiner sind als ein Byte, ist der Datentransfer auf das nächste ganzzahlige Vielfache auszuweiten, wobei die MSBs, die die Differenz zur Registerlänge sind, don't-care Zustände darstellen.

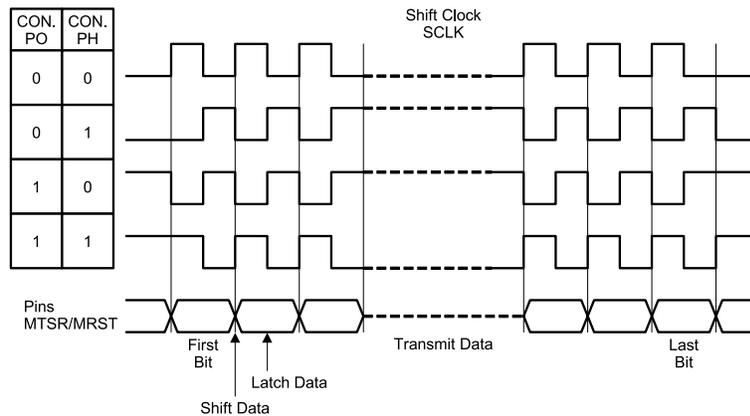


Abbildung 3.14: Darstellung der Betriebszustände [V1.04b].

Durch eine fallende Flanke an der CS Leitung wird der ADE7758 in den Kommunikationsmodus versetzt bzw. das serielle Interface rückgesetzt. Als erste Kommunikation die stattfindet, erwartet der ADE7758 das Schreiben eines Bytes ins Kommunikationsregister, welches den seriellen Datentransfer kontrolliert. Das erste Byte, das der Mikrocontroller nach der fallenden Flanke von CS über die SPI-Schnittstelle ausgibt, wird automatisch in dieses 8 Bit lange Register übernommen. Das MSB dieses Bytes gibt an, ob als nächste Operation ein Lesen, bei einer logischen Null, oder ein Schreiben, bei einer logischen Eins, auf die von den folgenden sieben LSBs spezifizierte Adresse stattfindet. In diesem Sinne ist die erste Art von Kommunikation, die stattfindet, immer ein Schreiben. Die Abbildung 3.15 veranschaulicht die Kommunikationsstruktur. Da es sich um einen synchronen

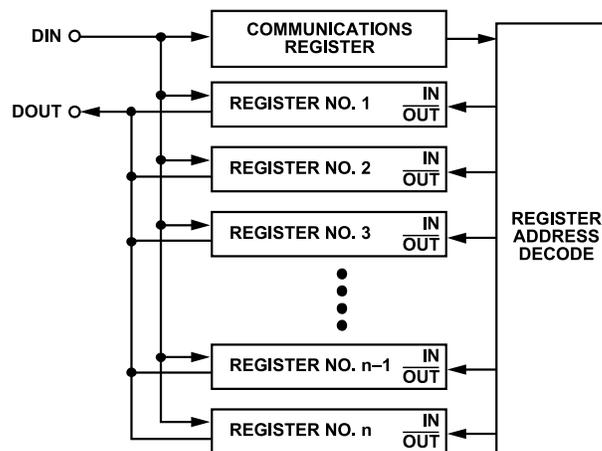


Abbildung 3.15: ADE7758 Kommunikationsstruktur [ADE06].

Bus handelt und Daten zur gleichen Zeit gelesen und geschrieben werden, wird beim Lesen von Bytes aus bestimmten Registern des ADE7758 zugleich auch immer ein Dummy-Byte geschrieben, selbst wenn es nur 0x00 ist.

Die Abbildung des Schreibzyklus 3.16 zeigt einen Schreibzugriff auf den ADE7758, das Senden des Command Bytes und der Byte-weisen Daten, die logische Eins des MSBs im Command-Byte, die das Schreiben in das ausgewählte Register signalisiert, sowie die zu

beachtenden Zeiten, die dem Datenblatt zu entnehmen sind.

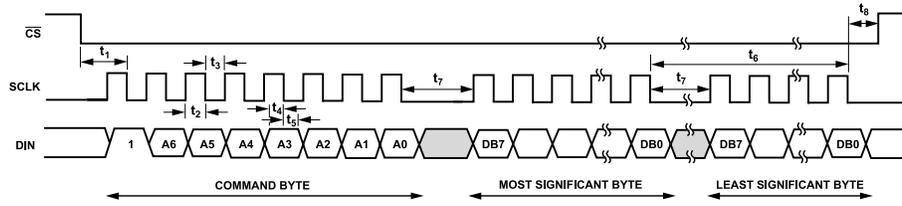


Abbildung 3.16: ADE7758 Schreibzyklus [ADE06].

Die Abbildung des Lesezyklus 3.17 veranschaulicht den Lesezugriff, der aufgrund der logischen Null des MSB's im gesendeten Command-Bytes an der ausgewählten Adresse durchgeführt wird, sowie die Byte-weise ausgelesenen Daten. Besonders ist auf das Timing bei schnellen Zugriffen im Zusammenhang mit Leseoperationen zu achten. Die minimale Zeit zwischen dem Read-Kommando, was einem Schreiben ins Communication-Register entspricht (letzte fallende Flanke von SCLK beim Schreiben) und der ersten steigenden Flanke der Clock beim Lesen des ersten Bytes des betroffenen Registers beträgt mindestens $4\mu s$.

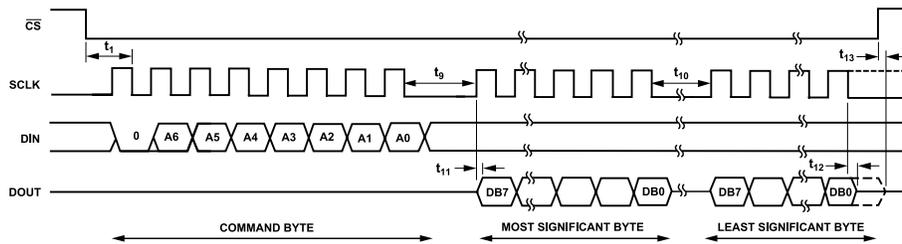


Abbildung 3.17: ADE7758 Lesezyklus [ADE06].

3.2 Anpassungsnetzwerk und Präzisionsspannungsteiler

Die Leiterspannung in der Sterntopologie bzw. die Strangspannungen in der Dreieckstopologie werden durch einen Spannungsteiler, bestehend aus Präzisionswiderständen auf den, vom ADE7758 geforderten Eingangsspannungsbereich von $+0.5V$ bis $-0.5V$ abgebildet. Der Spannungsteiler wird aus zwei $619k\Omega$ Widerständen und einem $1k\Omega$ Widerstand aufgebaut, an dem die Spannung abfällt, die vom ADE7758 gemessen wird (siehe dazu Schaltplan im Anhang). Diese Widerstände haben eine Toleranz von $\pm 0,1\%$ und einen Temperaturkoeffizienten von $15ppm/^\circ C$, was nötig ist, um dem Messgerät eine gewisse Langzeitstabilität, nach erfolgter Justierung und den damit verbundenen Abgleich der Toleranzen zu verleihen. Nicht zu vernachlässigen ist auch die Grenzspannung der Widerstände, bei hochohmigen Werten mit geringer Verlustleistung, die die Belastbarkeitsgrenze darstellt und nicht, auch nur impulsweise, überschritten werden darf. Die vorliegenden Widerstände haben eine maximal zulässige Spannung von $350V$ pro Element. Werden zwei davon in Serie geschaltet ergeben sich somit $700V$, die daran anliegen dürfen, was bei der maximal anliegenden Außenleiterspannung von $400V_{eff}$ noch 23% Reserve bis zum Erreichen des

maximalen Spitzenwertes ergibt. Einen zusätzlichen Schutz bietet ein Varistor mit einer geringeren Schwellspannung.

Die Nennbelastbarkeit der Widerstände beträgt bei 70°C 0.25W und ist somit bei niedrigen Strömen, bedingt durch den hochohmigen Gesamtwiderstand des Eingangsspannungsteilers, ausreichend. Die Grenzfrequenz und der Phasengang des, an den Eingangleitungen durch den Kondensator des Aliasingfilters und den Spannungsteiler gebildeten Netzwerkes sollte annähernd gleich der Grenzfrequenz und dem Phasengang des RC Aliasing Filter im Strompfad sein.

Der Strom wird durch drei HX 05 - NP Hallensoren, die abhängig von der Verbindung ihre Pins im Primärstromkreis einen Eingangsbereich I_{PN} von entweder 5A oder 10A Effektivwert aufweisen, in einen Ausgangsspannungsbereich ($R_L = 10\text{k}\Omega$) von ± 4 Volt Effektivwert abgebildet. Die Fehlergrenzen des Ausgangsspannungssignals, zwischen denen der wahre Wert liegt, beträgt bei einem Strom I_{PN} und einer Temperatur von 25°C typischerweise $\pm 1\%$. Der Hall Sensor arbeitet nach dem Open Loop Prinzip (Abbildung 3.18).

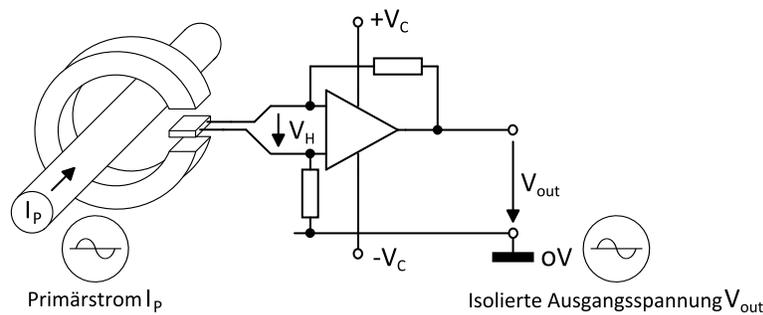


Abbildung 3.18: Open Loop Hall Stromwandler

Der Strom durch den Leiter bewirkt proportional zu seiner Stärke ein Magnetfeld, dessen Fluss im Magnetkreis verdichtet wird, und am Hall Sensor, durch Ablenkung der Ladungsträger in Folge der Lorentzkraft eine Spannung proportional dazu erzeugt. Sie ist bei einem konstanten Strom durch das Hallelement von der Stärke und Richtung des magnetischen Feldes abhängig und wird, wie in Abbildung 3.18 veranschaulicht, verstärkt und so aufbereitet, dass das Spannungssignal am Ausgang ein exaktes Abbild des Stroms durch den Leiter ist.

Durch eine serielle Verbindung der Pins der Primärseite wird der Primärstrombereich auf bis zu 5A festgelegt. Diese Stromhöhe macht eine entsprechend breite Verbindung nötig, um die Leiterbahn nicht zu stark zu erwärmen. Die Strombelastbarkeit ist abhängig von der Schichtdicke der Platine, von der Leiterbreite und der Kühlung durch das Substrat der Platine bzw. der Umgebung oder Luftbewegung.

Um auch Ströme höherer Größenordnung zu messen, kann ein Stromwandler mit 5A sekundärseitigen Ausgangsstrom bei primärseitigem Nennstrom und einer guten Genauigkeitsklasse ($0.1; 0.2; 0.5$ Fehler in Prozent bei primärem Bemessungsstrom und sekundärseitiger Bemessungsbürde) vorgeschaltet werden. Dadurch kann eine zusätzliche Phasenverschiebung entstehen, die sich bei der Energie- und Leistungsberechnung bemerkbar macht. Der Ausgang der Stromwandler stellt eine Konstantstromquelle dar. Bei zunehmender Bürde (Scheinwiderstand der sekundärseitig angeschlossenen Geräte einschließlich Zulei-

tungen) erhöht sich daher die Ausgangsspannung (entsprechend der Beziehung $U = R \cdot I$) so lange, bis Sättigung erreicht wird. Oberhalb der Sättigung steigt die Spannung bei zunehmender Verzerrung weiter an und erreicht ihren Maximalwert bei unendlich großer Bürde, also offenen Sekundärklemmen. Bei Stromfluss in der Primärwicklung darf die Bürde, die niedrig gehalten werden sollte und einen gewissen Wert (Bemessungsbürde) nicht überschreiten darf, damit die Spannung klein bleibt (niederohmige Messgeräte), erst abgeklemmt werden, wenn vorher die Sekundärseite kurzgeschlossen wurde. Die Sekundärseite muss als Schutz vor Überspannungen sekundärseitig geerdet werden. Bei der reinen Strommessung spielt die Polung des Wandlers keine Rolle, doch zur Ermittlung der Energieflussrichtung und bei einer rechnerischen Verknüpfung von Strom und Spannung wie bei der Leistungsberechnung, ist die Beachtung der Polarität jedoch unerlässlich.

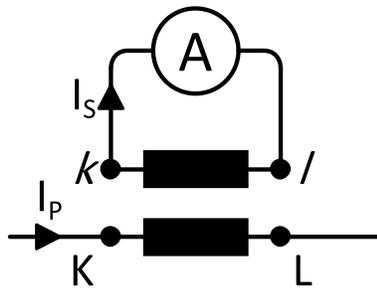


Abbildung 3.19: Schaltung eines Stromwandlers

So kann das Gerät einerseits auf den gewünschten Messbereich, durch Austausch der Wandler angepasst werden, andererseits muss, der Strompfad durch den Einsatz von klappbaren Wandlern nicht durchtrennt werden. Selbiges könnte man natürlich gleich durch den Einsatz klappbarer Hallsensoren realisieren, doch ist dies zu teuer und der Einsatz einer flexiblen klappbaren Rogowski Spule aufgrund der Tatsache von elektromagnetischer Sensibilität nicht erste Wahl. Abgesehen von den schon erwähnten Tatsachen, eine kleine Zusammenstellung der Vor- und Nachteile von Stromsensoren findet sich in Tabelle 3.2.

Der Ausgangsspannungsbereich von $\pm 4V$ der Hallsensoren muss ebenfalls durch einen Präzisionsspannungsteiler auf den Eingangsbereich von $\pm 0.5V$ der differenziellen Eingänge des Strompfades gebracht werden. Die $\pm 4V$ am Ausgang stellen nun aber den Effektivwert dar, worauf bei der Dimensionierung des Spannungsteilers geachtet werden muss, genauso wie auf die Eingangsimpedanz des Messkreises ($R_L = 10k\Omega$). Der aus den Widerständen mit den Werten 953Ω und $10k\Omega$ gebildete Teiler ermöglicht bei einem maximalen Eingangsstrom von $5A_{eff}$ eine Spannung von $0.348V_{eff}$ oder ± 0.492 an den differenziellen Eingängen des Strompfades und damit eine bestmögliche Ausnutzung des verfügbaren Messbereichs des ADE7758 (siehe dazu Schaltplan im Anhang). Neben dem maximalen Signalpegel am Eingang ist auch die zulässige Common-Mode Spannung zu beachten. Als Mittelwert der beiden Eingangsspannungen gegenüber dem Bezugspotenzial wird dadurch die Schaltung der Teilerwiderstände gegenüber Masse bestimmt. Wird die erlaubte Common-Mode Spannung überschritten, stellt der ADE7758 seine Funktion ein. Leichte Phasenunterschiede von Strom und Spannung die durch einen geringfügig abweichenden Phasengang der Eingangsnetzwerke für Strom und Spannung hervorgerufen werden können durch eine Verzögerung oder Beschleunigung der Signale intern kompensiert werden.

Anstelle des Stromwandlers könnte auch ein niederohmiger Shunt zum Messen des

Sensor	Vorteile	Nachteile
Niederohmiger Shunt	Geringer Preis Gute Linearität	Geringe Strombelastbarkeit DC Offset Parasitäre Induktivität
Stromwandler	Hohe Strombelastbarkeit Geringe Leistungsaufnahme	Hysterese / Sättigung Phasenverschiebung
Hall Effekt Sensor	Hohe Strombelastbarkeit Großer dynamischer Bereich	Hysterese / Sättigung Hohe Kosten Temperaturdrift
Ragowski Spule	Geringe Kosten Kein Sättigungslimit Geringe Leistungsaufnahme DC Offset Immunität Großer dynamischer Bereich Geringer Temperaturbereich	EMV Sensibilität Integration erforderlich

Tabelle 3.2: Vergleich Stromsensoren

Eingangstromes mit den in der Tabelle angeführten Nachteilen eingesetzt werden. Die daran abfallende Spannung müsste durch einen Impedanzwandler, mit hochohmigem Eingang und niederohmigen Ausgang von dem nachfolgenden Anpassungsnetzwerk getrennt sein. Der Hallsensor bietet aber den Vorteil der sicheren galvanischen Trennung zwischen Primär- und Sekundärkreis.

3.3 Antialiasing Filter

Dieser Filter verhindert Aliasing Effekte, die in jedem abtastenden System auftauchen, das analoge Signale digitalisiert. Wenn Eingangssignale Frequenzkomponenten enthalten die höher sind als die halbe Abtastrate der ADCs, so ist das Abtasttheorem ($f_{abtast} > 2 * f_{max}$) verletzt und diese Frequenzanteile werden als niedrigere Frequenzen interpretiert. Gerade die hohen Frequenzen nahe der Abtastfrequenz werden so in den Frequenzbereich der von Interesse für die Messungen ist (40Hz - 2kHz)gespiegelt (vgl. Abbildung 3.20).

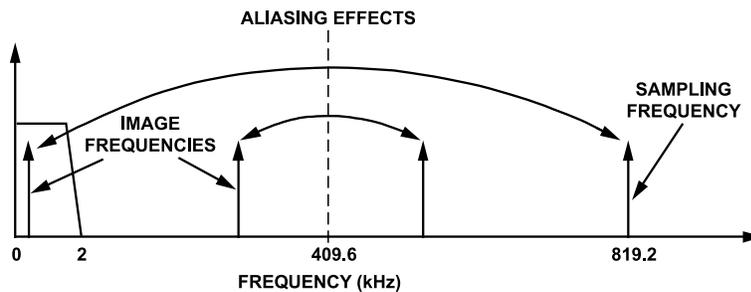


Abbildung 3.20: Aliasing Effekt [ADE07].

Die Kombination aus hoch abtastenden ADCs im ADE7758 (ungefähr 800kHz) und der geringen Bandbreite des Energy Meters (14kHz) ermöglicht die Anwendung eines simplen

Tiefpasses als Aliasingfilter. Eine ausreichende Filterung würde man mit einem RC Filter mit einer Grenzfrequenz von 10kHz erzielen, was bei einem Tiefpass 1. Ordnung und einer Verstärkung von -20dB pro Dekade, im Bereich der Abtastfrequenz eine Abschwächung von ungefähr 40dB oder 1/100 des Eingangssignals entspricht. Realisiert wurde dieses Aliasing RC Filter an jeder Analogleitung mit einem $1k\Omega$ Präzisionswiderstand und einem 33nF (10% X7R) Kondensator, woraus eine Grenzfrequenz von 4.8kHz folgt (siehe dazu Schaltplan im Anhang). Die Filter im Signalpfad für Spannung und Strom sind in Abhängigkeit voneinander zu entwerfen, da der Amplituden und Phasengang gegenseitig angepasst werden muss. Fehlanpassungen von Spannungs- und Stromzweig führen aufgrund unterschiedlicher Phasenverläufe der Filter zu Phasenfehlern bei der Messung von Energie und Leistung. Der dadurch entstandene Fehler in Prozent kann durch folgende Formel ausgedrückt werden: $Error = Mismatch(rad) * \tan(\theta) * 100\%$. Der Phasenwinkel zwischen dem gemessenen Strom- und Spannungssignal wird darin als θ bezeichnet. Daraus folgt, dass schon kleine Phasenverschiebungen, begründet durch das EingangsfILTER, insbesondere bei großem Phasenwinkel der Eingangsgrößen und damit kleinen Powerfaktoren, größere Fehler in der Messung verursachen. Phasenverschiebungen im geringeren Ausmaß können wie im Kapitel Phasenkalibrierung beschrieben, durch internes Verzögern oder Beschleunigen des Signals abgeglichen werden. Die Kondensatoren sollten so nahe aneinander wie möglich platziert und kürzestmögliche Abstände zu den Eingangspins des ADE7758 aufweisen, um möglichst gleiche Verhältnisse zu bekommen und die wirksame Länge der ungeschützten Anschlussleitungen möglichst kurz zu halten.

3.4 Messbereich

Die Messbereichsendwerte für Spannung und Strom werden durch das Anpassungsverhältnis der Eingangsnetzwerke bestimmt, deren Aufgabe es ist, das Niveau der Messgröße auf ein abbildendes Spannungsniveau zu bringen, wobei die maximale Spannung an den analogen Eingängen auf $\pm 500m\hat{V}$ beschränkt ist. Eine untere Grenze, bis zu der die Messung noch zuverlässige Werte liefert, besteht durch den linearen Bereich der Effektivwertmessung, in welchem der Fehler typischerweise 0,5% vom Endwert beträgt und der sich für Strommessungen bis zu 1/500 und für Spannungsmessungen bis zu 1/20 des Bereichsendwertes erstreckt.

Um den Messbereich des Hallsensors bis zum Nennwert des Primärstroms voll auszunutzen, sollte die Obergrenze direkt auf die des Energy Metering IC's übersetzt werden. Bei der Dimensionierung des Spannungsteilers am Ausgang des Hallsensors mit Widerstandswerten von 953Ω und $10k\Omega$ ergibt sich ein solches Übersetzungsverhältnis. Der maximale Strom, der somit gemessen werden kann, beläuft sich auf 5Aeff. Größere Ströme haben Ausgangsspannungen zur Folge, die die Spannungsschwelle der analogen Eingänge überschreiten und somit nicht mehr akkurat gemessen werden können. Zusätzlich wird auch der Fehler des Hallsensors aufgrund von Sättigung größer. Die maximale Spannung, die an den Analogeingängen sowohl für Spannung als auch Strom anliegen darf, ist $\pm 6\hat{V}$. Ein Überschreiten dieses Wertes kann die Beschädigung des IC's zur Folge haben. Ausgehend vom angegebenen linearen Bereich und vorliegender Eingangsschaltung folgt als untere Grenze des Messbereichs für Strom ca. 10mAeff. Doch ergibt ein derart niedriger Strom eine geringe Spannung am Ausgang des Hall Sensors, die durch Rauschen und Störungen

überlagert ist und somit wenig Aussagekraft besitzt. Deutlich wird dies auch unter Beachtung des Fehlers, der bei der Effektivwertmessung 0,5% des Skalenendwertes beträgt und in die Größenordnung der niedrigen Eingangsspannung fällt.

Der lineare Spannungsmessbereich erstreckt sich, bei der Wahl von zwei Widerständen in Serie, mit einem jeweiligen Wert von $619k\Omega$, zu einem $1k\Omega$ Widerstand, an dem die auftretende Spannung gemessen wird, von ca. 21.9V bis zu einem Skalenendwert von 438V. Um noch eine Reserve hinsichtlich der Grenzspannung der Teilerwiderstände zu haben, die die absolute Belastbarkeitsgrenze darstellt, sind Messungen höherer Spannungen zu vermeiden. Bei Messungen am 400V Drei-Phasennetz in Sternschaltung, also bei 230V gegenüber Erde, erreicht der Wert der Eingangsspannung des ADE7758 52.5% und bei Dreieckschaltung und einer Messung der Strangspannung von 400V ca. 91.3% vom möglichen Fullscale Wert.

3.5 Kalibrierung und Justierung

Unter Kalibrierung eines Gerätes versteht man allgemein den Vergleich eines damit ermittelten Wertes mit einer eindeutigen Referenzgröße oder einem Referenzgerät, unter bekannten Bedingungen zur Bestimmung der Abweichung. Justierung bezeichnet den Prozess der Minimierung der Abweichung durch Korrektur dieser Werte zur Steigerung der Genauigkeit.

Alle externen Elemente des Messzweiges wie Stromwandler, Spannungsteiler als auch die Internen wie Spannungsreferenz, A/D Umsetzer und Filter sind Einflussgrößen, die sich auf das Messergebnis auswirken. Mit der Kalibrierung soll ein bestmöglicher Zusammenhang zwischen der digitalen Ausgangsgröße und der in analoger Form vorliegenden Eingangsgröße und deren repräsentierten Messgröße unter Berücksichtigung der statistischen und symmetrischen Fehler hergestellt werden. Man bekommt einerseits die Beziehung zwischen den Bits der Register und deren entsprechenden physikalischen Einheiten, sowie erwirkt auch die Kompensation von Schwankungsdifferenzen der Phasen untereinander aufgrund von Bauteilvariationen im Eingangsnetzwerk.

Der Vorgang der Kalibrierung setzt das Wissen um die Übertragungskennlinie des Gesamtsystems sowie der dazugehörigen wahren oder richtigen Messgrößen voraus. Die Umsetzerkennlinie eines A/D-Systems erhält man durch Aufnahme zweier oder mehrerer Punkte von dem bekannten analogen Werten und dem dabei ausgegebenen Digitalwerten. Damit lassen sich dann Aussagen über den Nullpunktfehler, die Verstärkung und bei Aufnahme mehrerer Punkte über die Linearität des Systems machen. Bei der Aufnahme von lediglich zwei Punkten sollten diese nahe an den Bereichsendgrenzen der analogen Werte liegen, um damit den gesamten Messbereich charakterisieren zu können. Unter dem Begriff Umsetzerkennlinie wird hier nicht der direkte Zusammenhang zwischen analogem Wert und digitalem Wert direkt am Ausgang des A/D Wandlers verstanden, sondern die Beziehung zwischen analogem Wert und dem digitalen Wert in dem entsprechenden Register nach erfolgter Signalverarbeitung.

Während des Kalibrierungsprozesses benötigt man also eine stabile Eingangsgröße über den gesamten Messbereich und die Kenntnis über ihren genauen Wert. Für die Netzanalyse sind diese Eingangsgrößen stabile Spannung und Strom bei konstanter Frequenz an einer gleichmäßigen Last. Nimmt man die Frequenz mit konstanten 50Hz an, müssen Spannung

und Strom mit einem Multimeter möglichst exakt ermittelt werden. Letzten Endes hängt die Genauigkeit des Gesamtsystems von der Genauigkeit der einzelnen Komponenten und dem Aufwand bei der Kalibrierung ab.

Die Ausgabe der jeweiligen Digitalwerte im Kalibrierprozess erfolgt über die serielle Schnittstelle und wird am Computer in einem Terminalprogramm aufgezeichnet. Aufgrund der erwähnten Schwankungen werden bei einem stabilen Analogwert mehrere Digitalwerte aufgenommen, die dann in ein Tabellenkalkulationsprogramm importiert und deren Mittelwerte bestimmt werden können. Mit Bestimmung der Standardabweichung, als Maß der Streuung mehrerer Werte um ihren Mittelwert lässt sich eine Abschätzung über die Verteilung der ermittelten Digitalwerte machen und mit einer gewissen Wahrscheinlichkeit, mit der der richtige Wert im Schwankungsbereich liegt, Ausreißer ausgrenzen.

3.5.1 Kalibriergrundlage

Laut Datenblatt beträgt der Fehler bei Effektivwertmessung von Strom und Spannung 0.5% vom jeweiligen Messbereichsendwert über einen Messbereich von 500:1 bei Strommessung und 20:1 bei Spannungsmessung. Für den durch den Hallsensor und den nachfolgenden Spannungsteiler bestimmten Bereichsendwerts von 5A folgt damit ein typischer Fehler von 25mA. Aus dem Endwert des Messbereichs der Spannung von ca. 438 Volt errechnet sich folglich ein Fehler von 2.19V. Unter Berücksichtigung der Fehler des Hallsensors und den Ungenauigkeiten des Eingangsnetzwerkes, aufgrund von Toleranzen, sowie der Netzschwankungen, kann bestenfalls eine Abschätzung der Genauigkeit des vorliegenden Gerätes gemacht werden, die in die Größenordnung der bekannten Fehler fällt, unter Beachtung der Unsicherheit der Multimeter, die zur Kalibrierung als Referenzmessgeräte herangezogen werden.

Zur Referenzmessung von Spannung und Strom wurden zwei Multimeter des Typs 72-6202 der Firma Tenma verwendet. Die Genauigkeit dieser Geräte liegt bei (0.7% + 5 Digits) bei einer Wechselspannungsmessung und (1.0% + 5 Digits) bei einer AC Strommessung jeweils bei 50Hz Netzfrequenz. Zur Kalibrierung, unter den Bedingungen von möglichst konstanter Spannung und Strom, wurde ein am Institut für Elektronik vorhandener Netzsimulator der Firma Spitzenberger + Spies Modell Power Amplifier EM400/B mit 400 Watt Ausgangsleistung benutzt. Die Amplitude der rein sinusförmigen Spannung lässt sich im Promillebereich der maximalen Ausgangsspannung von 270V verstellen. Ebenso kann die Frequenz im entsprechenden Rahmen frei gewählt werden. Die maximale Ausgangsleistung sowie die Ausgangsspannung beschränken den Wertebereich, in welchem somit eine Kalibrierung durchführbar ist.

Für die Kalibrierung ist es vorteilhaft, diese mit einer einzigen Phase, angeschlossen an allen drei Spannungseingängen und mit einer Reihenschaltung der Hallsensoren durchzuführen, um auch die Anpassung der einzelnen Phasen untereinander, aufgrund der vorherrschenden gleichen Verhältnisse an den Eingängen bewerten zu können. In der Praxis stellt sich heraus, dass Schwankungen des Netzes die genaue Kalibrierung über den gesamten Messbereich erschweren. Dies gilt für die Spannung als auch für den lastbedingten Strom und damit insbesondere für die resultierende Leistung als Multiplikation der beiden Größen. Diese Multiplikation zweier mit einer gewissen Unsicherheit in Prozent behafteter Größen resultiert in einem Fehler, bestehend aus der Summe der Prozentwerte vom Multiplikationswert plus den hundertsten Teil der Multiplikation der Prozentwerte

vom Multiplikationswert. Durch die in den Signalpfaden eingesetzten Tiefpässen und der daraus resultierenden Dämpfung der Signale sind die ermittelten Kalibrierungskoeffizienten nur bei der für die Kalibrierung verwendeten Frequenz genau. Dies macht eine separate Kalibrierung sowohl bei 50 Hz als auch bei 60 Hz erforderlich. Dem Mikrocontroller muss darüber hinaus anhand einer Frequenzmessung durch den ADE7758 entscheiden, welche Kalibrierungsparameter zu laden sind. Eine eigenständige Kalibrierung zur Reduzierung des Offset ist neben dem Deaktivieren des HPF im Strompfad auch bei der Messung von Gleichgrößen nötig.

3.5.2 Beschreibung Umsetzerkennlinie

Charakteristisch für eine reale Umsetzerkennlinie sind die Abweichungen von ihrer idealen Form. Diese Abweichungen bzw. Fehler müssen für eine anschließende Justierung bzw. Korrektur erfasst werden. Durch Aufnahme mehrerer Punkte der Umsetzerkennlinie lassen sich entsprechende Parameter für eine bestmögliche Beschreibung gewinnen. Es besteht nicht die Möglichkeit direkt auf den A/D-Wandler einzuwirken, da dieser die Abtastwerte des Eingangssinussignals direkt an die verschiedenen Zweige des Signalverarbeitungsteils übergibt. In diesen kann dann auf die Berechnung der jeweiligen Werte korrigierend eingegriffen werden und/oder im Berechnungsprozess direkt im Mikrocontroller. Typische Abweichungen vom idealen Umsetzverhalten sind:

Nullpunktfehler (Offset Error) Als Offset versteht man einen additiven Fehler des Ausgabewerts des A/D-Wandlers, der bei 0V am Eingang des A/D-Wandlers eine Ausgabe an dessen Ausgang verschieden von Null verursacht.

Verstärkungsfehler (Gain Error) Der Verstärkungsfehler eines jeden Analog-Digitalwandlers ist definiert als der Unterschied zwischen der realen und der idealen Eingangsspannung, bei der der A/D Umsetzer seinen Maximalwert ausgibt, unter der Voraussetzung eines abgeglichenen Offset Fehlers und wird in Prozent angegeben. Durch den Verstärkungsfehler kann der eigentliche Messbereich nicht voll ausgeschöpft werden, da der Digitalwert schon vorzeitig sein Maximum erreicht hat.

Linearitätsfehler Weichen die ausgegebenen Werte innerhalb des festgelegten Messbereiches von einer Geraden ab spricht man von einem Linearitätsfehler. Dieser ist nicht abgleichbar, das Endergebnis kann aber mathematisch korrigiert werden, wenn man viele Referenzpunkte zur Ermittlung der Umsetzerfunktion heranzieht und daraus entsprechende Koeffizienten gewinnen kann, um Aussagen über die Nichtlinearität zu machen.

3.5.3 Effektivwerte

Die Auflösung der Effektivwerte von Spannung und Strom beträgt 24 Bit. Durch nicht exakt stabile Eingangsgrößen bei der Kalibrierung und nicht ideale Tiefpässe in den Effektivwert-Messzweigen, die einen Ripple mit der doppelten Frequenz des Eingangssignals am ausgegebenen Effektivwert verursachen, kommt es zu Schwankungen selbiger. Durch das zeitgleiche Auslesen der Werte bei auftretenden Nulldurchgängen lassen sich die Auswirkungen dieses Ripples beim Lesen der Digitalwerte minimieren. Aufgrund der

großen Auflösung fällt diese Schwankungsbreite auf den analogen Eingangsbereich umgelegt doch ziemlich gering aus.

Die Analog-Digital-Umsetzerkennlinie des ADE7758 ist im Bereich nahe dem Nullpunkt, im Gegensatz zum angegebenen dynamischen Messbereich von 500:1 bei Effektivwertmessung von Strom und 20:1 bei Effektivwertmessung von Spannung, nicht linear. Aus der Beschaltung der Messeingänge und dem resultierenden Eingangsbereich von 438V und 5A folgen für den linearen Bereich eine untere Spannungsgrenze von 21.9V sowie eine untere Stromgrenze von 10mA. Wegen des großen linearen Messbereichs für Strom entspricht die untere Grenze für Stromwerte nach der Konfiguration des Hallsensors und dem Verhältnis des dazu gehörigen Ausgangsspannungsteilers einer Spannung von ca. 1mV am Eingang des A/D-Umsetzers und reicht damit in einen Bereich, der aufgrund von überlagernden Störungen für einen akkuraten Wert schwankungsanfällig ist. Für den Eingangsmessbereich niedrige Größen sind so von Rauschen überlagert und wenig repräsentativ. Auch sind eine exakte Offsetbestimmung und der Abgleich dieses Fehlers nur näherungsweise möglich.

Mit der Aufnahme zweier Punkte für Strom und Spannung unter Beachtung des Messbereichs und unter den stabilen Verhältnissen des Netzsimulators folgt die Umsetzerkennlinie, die im Prozess der Justierung um die Abweichungen von der idealen Kennlinie korrigiert wird.

Effektivwertoffset

In den Zweigen des Signalverarbeitungsteils besteht die Möglichkeit, den Offsetfehler der Effektivwertmessung von Strom und Spannung, durch die im ADE7758 vorgesehenen Offsetregister, mittels Addition eines entsprechenden im Vorzeichen umgedrehten Wertes zu kompensieren. Unter gleichen Verhältnissen an allen drei Eingängen für Spannung und Strom wurden die Registerwerte der Effektivwerte mehrfach bei auftretenden Nulldurchgängen ausgelesen und gemittelt und bei zwei verschiedenen, auseinander liegenden Eingangsgrößen die Koeffizienten der linearen Funktion ermittelt. Nimmt man mehrere Werte der Umsetzerkennlinie auf, ergibt sich ein nahezu linearer Verlauf, beachtet man die Größenordnung und die Schwankungsbreite der resultierenden Digitalwerte.

Aufgrund von Rauschen lässt sich der Offset, wie im Datenblatt angemerkt nicht auf Null kompensieren, wenn am Eingang Null anliegt bzw. dieser offen ist, weshalb der nutzbare Messbereich durch die erwähnten Schranken begrenzt ist und Werte darunter ignoriert werden sollten. Der Offset wird somit an dem Punkt, der errechneten unteren Bereichsgrenze ermittelt und nicht beim Nullpunkt, da von einer Nichtlinearität im unteren Messbereich auszugehen ist. Dies minimiert den bestehenden Offset bei offenem Eingang zwar nur gering, doch beschreibt die damit errechnete Umsetzerkennlinie den spezifizierten Messbereich genau. Dieser Schluss liegt nahe, da die Werte beim Vergleich der Scheinleistung, die aus der internen Multiplikation der Effektivwerte errechnet wird, mit dem Ergebnis der ermittelten Wirkleistung an einem ohmschen Widerstand gut übereinstimmen.

Der errechnete Offsetwert wird im Falle der Spannung durch 64 dividiert, da ein LSB der Spannungseffektivwertoffsetregister gleichbedeutend mit 64 LSBs des Spannungseffektivwertregisters ist, und entsprechend vorzeichenrichtig, damit sich der Offset aufhebt, ins Register eingesetzt. Im Falle des Stroms ist zu beachten, dass sich das Offsetregister aus Sicht des Signalfades vor dem Block der Wurzelbildung befindet und somit die Werte

im Rechengang quadriert eingesetzt werden müssen. Der ermittelte Offsetwert muss, da ein LSB der Stromeffektivwertoffsetregister gleichbedeutend mit 128 LSBs des Stromeffektivwertregisters ist, vor dem Einsetzen ins Offset ausgleichende Register, noch durch das Quadrat dieses Faktors dividiert werden. An dieser Stelle sei angemerkt, dass durch Kompensation des Offset der beiden Effektivwerte von Spannung und Strom gleichzeitig auch der Offset der Scheinleistung kompensiert sein sollte, da das Produkt der beiden Werte die Scheinleistung ergibt.

3.5.4 Regressionsrechnung MKQ

Mithilfe der Regressionsrechnung und dem Verfahren der Methode der kleinsten Quadrate (MKQ) lassen sich Zusammenhänge zwischen einer abhängigen Variablen, hier der Registerwert und einer Unabhängigen, hier die Eingangsgröße, die den Registerwert verursacht, erkennen und Aussagen über deren nicht deterministischen Verlauf treffen. Messergebnisse von Messreihen, die mit Fehlern überlagert sind, lassen sich so am exaktesten auswerten und eine Funktion ermitteln, die die Beziehung dieser Variablen bestmöglich beschreibt.

Der einfachste und erwünschte Fall bei einer Umsetzerkennlinie ist ein Polynom ersten Grades, das heißt, dass der Zusammenhang linear verläuft und einer Geraden entspricht. Zwei Werte, möglichst an den Messbereichsgrenzen genügen, um eine lineare Umsetzerkennlinie zu beschreiben, doch mehrere Messpunkte bei verschiedenen Eingangsgrößen und die Mittelung mehrerer ausgegebener Werte bei gleicher Eingangsgröße reduzieren die Streuung oder Standardabweichung. In der Realität liegen die aufgenommenen Punkte einer Messreihe meist nicht exakt auf einer Geraden, da die Digitalwerte von Störungen überlagert sind und die dazugehörigen Messwerte auch eine gewisse Schwankungsbreite haben. Mit der Methode der kleinsten Quadrate wird die Summe der Quadrate der vertikalen Abweichung zu den Punkten minimiert und so eine Funktion ermittelt, die den Verlauf bestmöglich beschreibt.

Wenn man den Zusammenhang zwischen x und y durch ein Polynom m -ten Grades bei n Messungen approximieren will, wobei die Anzahl der Messungen n größer sein muss als der Grad m bekommt man folgendes Gleichungssystem,

$$\begin{pmatrix} 1 & x_1 & \cdots & x_1^{m-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & \cdots & x_n^{m-1} \end{pmatrix} \cdot \begin{pmatrix} a_1 \\ \vdots \\ a_m \end{pmatrix} = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix} \quad (3.37)$$

dass nicht exakt gelöst werden kann. Die Lösung die den quadratischen Fehler minimiert errechnet sich aus

$$|Ma - y|^2 = \left(\sum_{k=1}^m a_k x_1^{k-1} - y_1 \right)^2 + \dots + \left(\sum_{k=1}^m a_k x_n^{k-1} - y_n \right)^2 \quad (3.38)$$

und ergibt

$$a^* = (M^T M)^{-1} M^T y. \quad (3.39)$$

Diese Berechnung der besten Koeffizienten der Ausgleichsfunktion ist mit MATLAB einfach durchzuführen und liefern bei steigender Anzahl an Messungen n , unter n verschiedenen Eingangswerten, einen exakteren Verlauf der Umsetzerkennlinie als bei der Aufnahme von lediglich zwei Werten, was sich auch in der Differenz zwischen den berechneten, ausgegebenen Werten und den tatsächlichen analogen Eingangswerten äußert.

3.5.5 Intervallweise lineare Approximation

Ist der Fehler über den gesamten Messbereich unterschiedlich groß, durch eine nicht optimale Anpassung der Umsetzerkennlinie an die ermittelten realen Werte, kann der gesamte Messbereich in mehrere kleine unterteilt werden. Für jeden dieser Teilbereiche werden dann die Steilheit und der Achsenabschnitt ermittelt. So wird die gesamte Umsetzerkennlinie durch mehrere kleiner lineare Kennlinien dargestellt, die intervallweise an die reale Kennlinie approximiert sind. Je größer die Anzahl der Vergleichsmesswerte über den Messbereich ist, umso kleiner werden die Abweichung von den realen Verhältnissen und damit der Fehler. Das Mikrocontrollerprogramm entscheidet anhand des gelesenen Digitalwertes, in welchem Intervall sich dieser befindet und berechnet anhand der Parameter für dieses den physikalisch richtigen Messwert.

3.5.6 Phasenkalibrierung

Eine Phasenverschiebung des Strom- und Spannungssignals, hervorgerufen durch einen Stromwandler mit typischen $0.1^\circ - 0.3^\circ$ Phasenfehler oder durch einen unterschiedlichen Phasengang des Eingangsnetzwerks resultiert in einem Fehler der Wirkleistungs- und Blindleistungsberechnung. Bei einem rein ohmschen Verbraucher mit Spannung und Strom in Phase verringert sich mit steigender Leistung die berechnete Wirkleistung im Verhältnis zur Tatsächlichen und es tritt geringe Blindleistung auf, die linear in gleichem Maße zunimmt.

Es ist möglich, im ADE7758 durch Einfügen einer Verzögerung oder Beschleunigung im Spannungszweig, das Signal im Zeitbereich von $-75.6\mu s$ bis $+153.6\mu s$ zu verschieben, was bei einem 50Hz Eingangssignal einer Verschiebung zwischen -1.36° und $+2.76^\circ$ entspricht. Jede Phase kann einzeln durch Setzen des zugehörigen Registers kalibriert werden, wobei ein LSB Zeitverzögerung $1.2\mu s$ und ein LSB Zeitbeschleunigung $2.4\mu s$ bedeutet. Positive Werte im Register haben eine Verzögerung der Spannung zur Folge und vice versa.

Betrachtet man die Eingangsnetzwerke des Messteils für Spannung und Strom und vergleicht diese miteinander, entsteht aufgrund der Tatsache, die Common-Mode Spannung im Strommesspfad zu minimieren und der damit getroffenen Wahl der Spannungsteiler am Ausgang des Hallsensors, eine theoretische Phasenverschiebung von -0.517° zum Spannungsmesspfad an den Analogeingängen, auch abhängig von den Toleranzen der verwendeten Bauteile. Dies entspricht bei 50Hz einer Verzögerung von $28.7\mu s$.

Vergleicht man an einer rein ohmschen Last, bei bereits durchgeführter Kalibrierung und Justierung der Effektivwertmessung, die folglich offsetfreie Scheinenergie mit der Wirkenergie, die ihrerseits über eine geringe Integrationsperiode zur Eliminierung eines Offset ermittelt wird, bei mehreren ansteigenden Leistungswerten, erkennt man einen eben solchen Anstieg der ausgegebenen Blindenergie. Es liegt also der Schluss einer Phasenverschiebung nahe. Berechnet wird der Phasenfehler in Prozent für jede Phase aus der

Differenz der Wirk- und Scheinleistung mit folgender Gleichung:

$$\text{Phasenfehler [\%]} = \frac{xWATTHR - xVAHR}{xVAHR} \cdot 100 \quad (3.40)$$

Bei mehreren Leistungswerten erkennt man einen annähernd konstanten Verlauf der Werte. Wenn 100% Phasenfehler bei 90° Phasenverschiebung auftreten, folgt daraus dann der Phasenfehler in Grad.

In der Praxis ergibt dieser Wert dann annähernd den, aufgrund des Frequenzgangs des Anpassungsnetzwerks, zuvor errechneten Wert für die Phasenverschiebung bei einer Netzfrequenz von 50Hz. Geht man von dem ermittelten theoretischen Wert aus, muss das voreilende Spannungssignal durch eine Verzögerung mittels Setzen des betroffenen Registers $xPHCAL$ korrigiert werden. Der Wert für eine Verzögerung bei 50Hz Netzfrequenz errechnet sich aus:

$$xPHCAL = \text{Phasenverschiebung [°]} \cdot \frac{1}{1.2\mu s} \cdot \frac{2000\mu s}{360^\circ} \quad (3.41)$$

Nimmt man diesen Wert für alle Phasen an, erkennt man anschließend eine deutliche Reduktion der gemessenen Blindleistung bei einem rein ohmschen Verbraucher. Aufgrund der Wertestreuung von Bauteilen kann eine leichte Korrektur von ein bis zwei LSB notwendig sein, um bei gleicher Belastung aller Phasen auch den gleichen geringen Blindleistungswert zu bekommen.

3.5.7 Energie bzw. Leistung

Es existieren im Signal-verarbeitenden Mess-IC keine Register, die den Wert der durchschnittlichen Leistung explizit beinhalten. Mit Kenntnis der Energie und der Integrationszeit kann aber auf die durchschnittliche Leistung dieser Zeitspanne zurückgerechnet werden. Da aber die Integrationszeit durch eine bestimmbare Zahl an Nulldurchgängen des Spannungssignals bestimmt wird, ist sie abhängig von der momentanen Netzfrequenz. Deren durchschnittlicher Wert während der Integration muss folglich mit aufgenommen werden. Durch Schwankungen und Ungenauigkeiten bei der Messung der Frequenz ergeben sich dadurch zusätzliche Fehler bei der Berechnung der durchschnittlichen Leistung.

Der Energiewert kann bei konstanter Netzfrequenz und damit genau bekannter Integrationsperiode, festgelegt durch die Anzahl an Nulldurchgängen, in Referenz zu einer mit anderen Messgeräten ermittelten Leistung gesetzt werden. Der Proportionalitätsfaktor zwischen der ermittelten Energie und der zu ermittelten mittleren Leistung ist dabei die Akkumulationszeit.

Fehler und Abweichungen

Laut Datenblatt beträgt der Fehler der Wirkenergiemessung typisch 0.1% vom Messbereichsendwert über einen dynamischen Bereich von 1000:1. Da die Energie die Integration der Leistung oder in einem digitalen System die Summe der Abtastwerte der Leistung über eine bestimmte Zeit ist, folgt für die Berechnung der durchschnittlichen Leistung für diese Zeit der gleiche Fehler in Prozent, bei bekannter Integrationsdauer. Der Messbereichsendwert der Leistung ergibt sich aus der Multiplikation, der durch die Anpassungsnetzwerke festgelegten Bereichsendwerte von Strom und Spannung und beläuft sich somit auf ca.

2190 Watt. Ein Fehler von 0.1% resultiert theoretisch somit in einer Unschärfe der Leistungsmessung von 2.19W über den gesamten Messbereich.

Bei der Wahl der Anzahl an Nulldurchgängen, über deren Zeitdauer die Leistung zu Energie aufsummiert werden soll, muss ein Kompromiss eingegangen werden zwischen der Länge der Akkumulierungszeit, sodass auch im unteren Messbereich akkurate Werte gelesen werden können und der Zeit bis zum Überlauf der Register bei Fullscale Input, unter Berücksichtigung der Tatsache, dass ein bestehender Offset am Multiplikatorausgang der Signalpfade für Strom und Spannung ebenfalls über die Integrationsdauer aufsummiert wird. Beobachtet man in der Praxis mehrere Werte der Energieregister bei offenen Stromeingängen, aber angelegter Spannung über verschiedene Integrationsperioden und Anzahl an Nulldurchgängen, erkennt man annähernd eine Normalverteilung dieser Werte, deren Standardabweichung um den Mittelwert Null mit steigender Integrationszeit zunimmt. Dies legt den Schluss nahe, dass das Messergebnis mit zufälligen Fehlern behaftet ist, die nicht korrigiert werden können, sondern nur durch Mittelung und vor allem durch Wahl einer geringen Integrationsperiode und damit geringer Streuung ausgeschlossen werden. Bei einem Offset als systematischen Fehler würde sich der Mittelwert der Messreihen bei größer werdender Integrationszeit auch systematisch verschieben. Da dies nicht der Fall ist, kann hier nicht von einem systematischen Offset gesprochen werden. Ein möglicher Offset der Leistungsmessung würde sich sowieso bei kleiner Integrationszeit nicht störend auf das Ergebnis auswirken, aufgrund der Eigenschaft, dass nur die obersten 16bit des 41bit breiten Summationsregisters gelesen werden.

Zusätzlich zu einer schwankenden Spannungsquelle und fehlerbehafteten Messgeräten zur Referenzbildung hat auch die Wahl eines geeigneten, rein ohmschen Lastwiderstandes, zur Bestimmung der daran umgesetzten Wirk- und Scheinenergie, Einfluss auf die Genauigkeit. Der Wert des Widerstandes sollte sich bei einem temperaturbedingten Drift schnell einem konstanten Wert annähern. Aufgrund der Landungsträgerbewegungen entsteht im Widerstand Wärme, die der umgesetzten Leistung entspricht. Bei größeren Strömen und damit größeren Leistungen verändert sich der Widerstand wegen seiner Temperaturabhängigkeit, was wiederum einen veränderten Stromfluss zur Folge hat. So kann eine scheinbar kleine Stromänderung bei Netzspannung schon eine Leistungsänderung in der Größenordnung von Watt nach sich ziehen. Im zur Kalibrierung verwendete Widerstand sollte sich daher schnell ein konstanter Nennstrom ausbilden, damit während der Messung keine Widerstandsänderungen sowie Stromänderungen auftreten. Die Parallelschaltung mehrerer Glühbirnen, die Kaltleiter sind und einen positiven Widerstandsverlauf haben, eignet sich daher gut, da durch den hohen Einschaltstrom schnell die Betriebstemperatur erreicht wird und sich das Gleichgewicht zwischen Strom und Widerstand einstellt.

Der Signalpfad der Leistungsberechnung verfügt zusätzlich über einen Multiplikator, der den Leistungswert durch Multiplikation mit einem Gain Wert im Bereich von $\pm 50\%$ des Leistungswertes anpassen kann. Der Nutzen besteht darin, die drei Phasen bei gleichen Verhältnissen an deren Eingängen trotz ihrer individuellen Fehler und Abweichungen voneinander, zu gleichen Ergebnissen von Digitalwerten in den Registern kommen zu lassen, um für die Berechnung der physikalischen Werte für alle Phasen die gleichen Koeffizienten heranziehen zu können. Dieses Gain Register besitzt dabei eine Auflösung von 0.024%/LSB, was jedoch angesichts der Werte im Leistungspfad nicht klein genug erscheint, um die Leistungswerte und damit die Ergebnisse aufeinander abzustimmen. Deswegen ist die Kalibrierung jeder einzelnen Phase durch Ermittlung der individuellen

Umsetzerkennlinie anhand der Parameter zur Berechnung der Ergebniswerte vorzuziehen. Mittels der DIV Register, die die durchschnittliche Leistung vor der Integration durch eine ganze Zahl teilen, wird die Zeit bis zu einem Überlauf des Akkumulationsregisters verlängert. Der Energiewert muss vor seiner Verarbeitung im Mikrocontroller mit eben dieser Zahl wieder multipliziert werden, bevor die Datenverarbeitung erfolgt.

Innenwiderstand

Bei der Messung zur Referenzbildung der mittleren Leistung bzw. Energie über die gewählte Anzahl an Halbwellen ist die Schaltung der Referenzmessgeräte von Bedeutung für die gemessenen Werte und damit für die Justierung und Kalibrierung des Geräts.

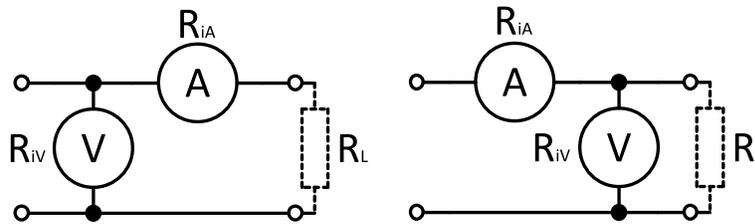


Abbildung 3.21: Strom- und Spannungsrichtige Schaltung Multimeter bzw. Wattmeter

Die Unterschiede werden deutlich, wenn man für die Messgeräte den Innenwiderstand im jeweiligen Messbereich betrachtet. Bei der stromrichtigen Schaltung fällt aufgrund des Innenwiderstandes des Amperemeters R_{iA} eine Spannung ab, wodurch die, mit dem Voltmeter ermittelte Spannung um eben diesen Betrag zu groß gemessen wird. Mit Kenntnis von R_{iA} kann das Ergebnis dementsprechend korrigiert werden. Bei der spannungsrichtigen Schaltung ist folglich die angezeigte Leistung um die am Innenwiderstand des Amperemeters verbrauchte Leistung größer als die Tatsächliche, am Verbraucher umgesetzte Leistung. Bei Kenntnis des Innenwiderstandes ergibt sich diese aus:

$$P = P_a - I^2 R_{iA} \quad \left\| \begin{array}{ll} P_a & \dots \text{ angezeigte Leistung} \\ R_{iA} & \dots \text{ Innenwiderstand Amperemeter} \end{array} \right.$$

Bei der spannungsrichtigen Messung wird infolge des Stroms, der wegen des endlichen Innenwiderstandes R_{iV} durch das Voltmeter fließt, ein zu großer Strom gemessen. Dabei ist die angezeigte Leistung als Produkt des gemessenen Stromwertes und des Spannungswertes um die verbrauchte Leistung am Innenwiderstand des Voltmeters zu hoch. Durch die Hochohmigkeit des Voltmeters kann hier auf eine Korrektur in der Regel verzichtet werden, da der Strom durch das Voltmeter sehr viel geringer als der Laststrom ausfällt. Die tatsächliche Leistung ergibt sich aus der gemessenen Leistung P_a aus:

$$P = P_a - \frac{U^2}{R_{iV}} \quad \left\| \begin{array}{ll} P_a & \dots \text{ angezeigte Leistung} \\ R_{iV} & \dots \text{ Innenwiderstand Voltmeter} \end{array} \right.$$

Wenn der Innenwiderstand des Voltmeters sehr viel größer ist als der zu messende Widerstand, kann also bei der spannungsrichtigen Messung auf eine Korrektur verzichtet werden. Bei hochohmigen Widerständen als Last ist hingegen eine stromrichtige Schaltung

zu bevorzugen, wenn der Spannungsabfall am Innenwiderstand des Amperemeters klein gegenüber dem, am Innenwiderstand des Voltmeters ist.

Die Schaltung, die einen geringeren Leistungsverbrauch durch das entsprechende Multimeter erfährt und damit die angezeigte Leistung im Vergleich zur anderen Schaltung geringer ist, ist bevorzugt einzusetzen, falls der Leistungsabfall nicht durch Rechnung korrigiert wird.

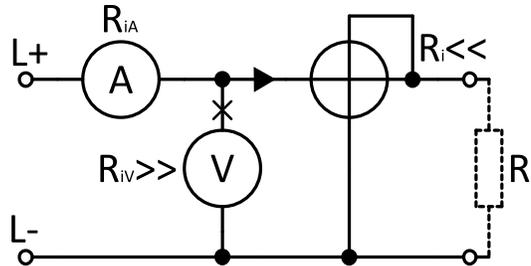


Abbildung 3.22: Spannungsrichtige Kalibrierschaltung

Durch einen Innenwiderstand des Spannungsmesszweiges im $M\Omega$ Bereich und durch die Erfassung des Stroms über Hallensoren, stellt das hier beschriebene Messgerät nahezu keinen Widerstand dar, womit die Leistungsaufnahme aus dem zu messenden Netz sehr gering ist und mit Null angenommen werden kann. Bei der Kalibrierung ist zu beachten, dass das zu kalibrierende Gerät, wenn man von einer Leistungsaufnahme durch dessen Messzweig von nahezu Null ausgeht, dem Lastwiderstand direkt vorgeschaltet wird und die Messgeräte zur Bestimmung der Referenzleistung, wie in der Abbildung 3.22 ersichtlich, dem zu kalibrierenden Gerät voranzustellen sind. Damit wird verhindert, dass die Leistungsaufnahme der Multimeter vom Messgerät mitgemessen wird und es so zu einer fehlerhaften Referenzbildung kommt. Als Schaltungstopologie wählt man die spannungsrichtige Schaltung da die zur Kalibrierung eingesetzte Last, beispielsweise Glühlampen mit großer Wirkleistungsaufnahme, einen sehr kleinen Widerstand gegenüber dem Innenwiderstand des Voltmeters darstellen.

Wirk- und Scheinenergie

An einer rein ohmschen Last entspricht die gemessene Scheinleistung der umgesetzten Wirkleistung. Beobachtet man dabei die Werte der zugehörigen Register für Schein- und Wirkenergie bei gleicher Integrationszeit nach durchgeführter Kalibrierung und Justierung am Energy Measurement IC, zeigt sich trotzdem ein geringer Unterschied der digitalen Werte, aufgrund der verschiedenartigen Struktur der Signalwege und deren internen Filtern. Diese Abweichung ist außerdem noch von der Netzfrequenz abhängig. Erst durch die entsprechende Korrektur der Werte im Mikrocontroller stimmen die Endergebnisse der Messwerte bestmöglich untereinander und mit den tatsächlichen physikalischen Werten überein.

Die Kalibrierung und Justierung der Schein- und Wirkleistung kann gemeinsam und simultan an einem rein ohmschen Widerstand durchgeführt werden. Dazu wird eine Phase mit dieser Last beaufschlagt, und alle Spannungseingänge daran parallel angeschlossen, als auch die Stromeingänge in Reihe verschalten, dass an allen Eingängen des Messgerätes die exakt gleichen Eingangsgrößen anliegen. Durch die stabile Frequenz des Netzsimulators ist

die Integrationszeit gegeben durch die gewählte Anzahl an Halbwellen, die an einer Phase erkannt und gezählt werden, eindeutig bestimmbar. Da so die Eingangszustände an allen Phasen die Gleichen sind, können mehrere korrelierende Messwerte der Energie und der durchschnittlichen Leistung während der Integrationszeit, über den Ausgangleistungsbereich des Netzsimulators (400W) aufgenommen und die Umsetzerkennlinie ermittelt werden.

Durch die Methode der kleinsten Quadrate bei Aufnahme mehrerer Wertepaare von Analog- und entsprechenden Digitalwerten lässt sich zeigen, dass sich der Fehler der Energieberechnung bei konstanter Netzfrequenz zwischen erwarteten Messwert und ausgegebenen Wert bei bestimmter Eingangsspannung durch eine Umsetzerkennlinie, die durch ein Polynom zweiten Grades beschrieben wird, minimieren lässt. Der Koeffizient, der in diesem Fall mit dem quadratischen Wert der Variablen multipliziert wird, ist sehr klein, wodurch der Graph der Funktion fast linear erscheint.

Blindenergie

Zur Kalibrierung und Justierung der Blindenergie bzw. Blindleistung könnte ein reiner Blindwiderstand herangezogen werden, damit die durch die Multimeter gemessene Scheinleistung der Blindleistung entspricht. Doch wird durch einen ohmschen Anteil das Ergebnis verfälscht und noch dazu muss ein Blindelement gefunden werden, das auch größere Blindleistungen, die zur Kalibrierung notwendig sind, verbrauchen kann. Beispielsweise sei hier eine Synchronmaschine erwähnt, die als Phasenschieber arbeitet.

Da die Blindleistung aber auch mit der arithmetischen Methode aus Kenntnis der Schein- und Wirkleistung berechnet werden kann, bieten sich außerdem an, diese im Mikrocontroller so direkt zu berechnen oder die kalibrierte Schein- und Wirkleistung, die durch das Messgerät erfasst wird, als Referenz an einer teilweisen induktiven Last wie einem Elektromotor zu gebrauchen. Dabei wird die Blindleistung arithmetisch berechnet und in Relation zu den Blindenergiewerten der entsprechenden Register des Energy Metering IC's gesetzt. Mit dieser Methode lässt sich die Umsetzerkennlinie, anhand der das Messgerät seine Ausgabewerte berechnet, über einen größeren Leistungsbereich ermitteln. Da die arithmetische Methode kein Vorzeichen der damit errechneten Blindenergie liefert, kann dieses parallel aus dem enthaltenen Wert des Energieregisters für Blindenergie abgeleitet werden.

3.5.8 Power Faktor

Die Genauigkeit des Powerfaktors, der sich aus den ermittelten Werten für Schein- und Wirkenergie ergibt, hängt von der Genauigkeit dieser Werte und deren Verhältnis zueinander ab. Betrachtet man den Powerfaktor als einen Wert mit zwei Nachkommastellen, folgt in der Praxis ein Fehler von etwa 1% bei durchgeführter Justierung der Effektivwertberechnung für Strom und Spannung und der korrigierten Phasenverschiebung sowie bei der Wahl einer geringen Integrationsperiode zur Energieberechnung, um zufällige Fehler im spezifizierten Messbereich zu minimieren.

3.6 Schaltungen zur Leistungsmessung

Im Dreiphasensystem wird die Leistung allgemein mit 3 Leistungsmessern aufgenommen. Bei symmetrischer Belastung genügt es aber praktisch, nur die Leistung eines einzigen Strangs zu messen, um auf die Gesamtleistung zu schließen, die sich aus dem 3fachen Einzelwert ergibt. Verbraucher können im Dreiphasennetz, das als Vierleiter- oder Dreileitersystem ausgelegt sein kann, einphasig oder in Stern- oder Dreieckschaltung angeschlossen sein. Daraus ergibt sich auch die Schaltung der Leistungsmessgeräte oder der Anschluss des hier vorliegenden dreiphasigen Energiezählers.

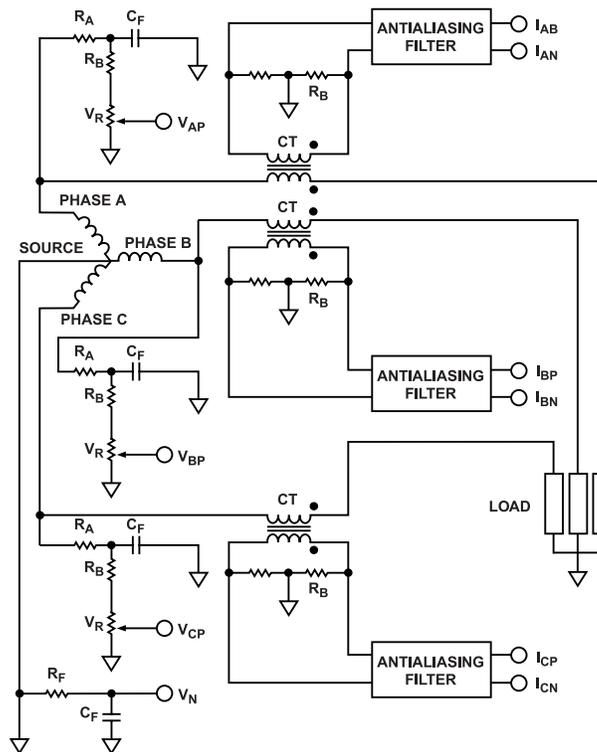


Abbildung 3.23: Anschluß Sternschaltung[Kap03].

Im Vierleiternetz werden der Sternpunkt und der daran angeschlossene Neutralleiter als Bezugspunkt der Messung herangezogen, wobei die Einzelleistungen der Phasen erfasst werden. Deren Summe ergibt die Gesamtleistung. Der Neutralleiter ist als aktiver Leiter des Netzes bei unsymmetrischer Belastung von den Ausgleichsströmen der Außenleiter durchflossen. Die Abbildung 3.23 veranschaulicht die entsprechende Anschlußkonfiguration.

Im Dreileiternetz ohne Neutralleiter (Dreieckschaltung, Sternschaltung ohne Nullleiter im Sternpunkt) ist die Bestimmung der Gesamtleistung bei symmetrischer und unsymmetrischer Last mit der Aronschaltung 3.25 möglich. Man erhält jedoch nicht die Einzelleistung der Phasen. Formal wird dabei ein Leiter als Rückleiter aufgefasst. Der Strom zweier Leiter, sowie deren Spannung gegenüber dem angenommenen Rückleiter (Außenleiterspannung) werden zur Ermittlung der Gesamtleistung gemessen.

Im Vierleiternetz mit Neutralleiter kann die Aronschaltung nur bei symmetrischer

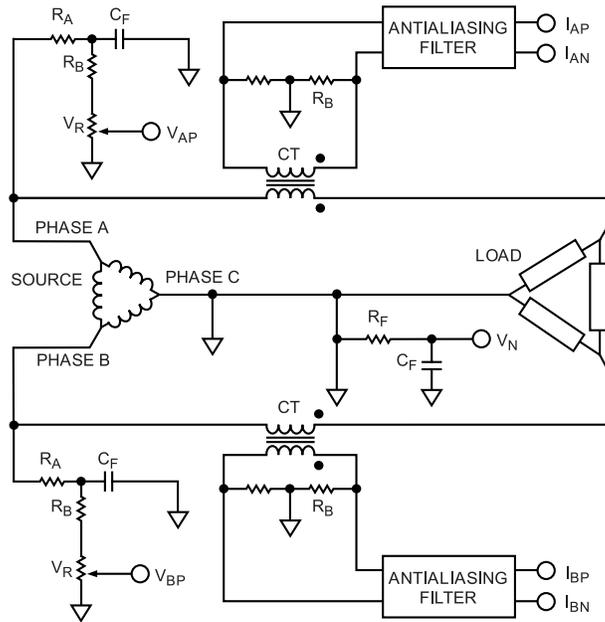


Abbildung 3.24: Anschluß Dreieckschaltung[Kap03].

Belastung eingesetzt werden, da nur dann der Strom im Neutralleiter I_N zu Null wird. Um das vorliegende Messgerät in der Aronschaltung zu betreiben, wird ein Leiter an den Spannungsbezugsanschluss (V_N) befestigt.

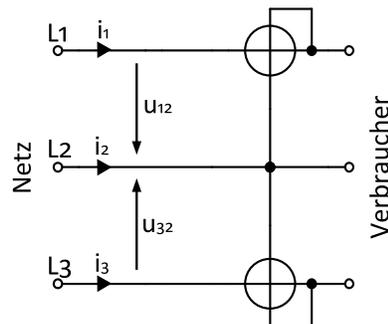


Abbildung 3.25: Aronschaltung

$$u_{12} = u_{1N} - u_{2N} \quad (3.42)$$

$$u_{32} = u_{3N} - u_{2N} \quad (3.43)$$

$$u_{1N} = u_{12} + u_{2N} \quad (3.44)$$

$$u_{3N} = u_{32} + u_{2N} \quad (3.45)$$

Daraus folgt weiter für p :

$$p = u_{1N}i_1 + u_{2N}i_2 + u_{3N}i_3 \quad (3.46)$$

$$= u_{12}i_1 + u_{32}i_3 + u_{2N}(i_1 + i_2 + i_3). \quad (3.47)$$

Im Dreileiterstromkreis gilt durch den fehlenden Neutralleiter laut Kirchhoff und unter der Annahme, dass keine Leckströme auftreten:

$$i_1 + i_2 + i_3 = 0 \quad (3.48)$$

Womit sich die Gesamtleistung des Dreileiternetzes sowohl bei symmetrischer als auch unsymmetrischer Belastung ergibt:

$$P_g = P_1 + P_3 \quad (3.49)$$

Kapitel 4

Netzwerk und Internetanbindung

Ausgehend von der Betrachtung der nötigen Hardware für die Anbindung eines eingebetteten Systems an ein Netzwerk und deren Zuordnung zum OSI-Schichtmodell bzw. zum Internet Referenzmodell soll hier über die Realisierung eines Netzwerk- bzw. Internetknotens und dessen wesentlichen Aspekten im Zusammenhang mit dieser Arbeit eingegangen werden. Ziel ist es einen Überblick über einen Embedded Webserver, seine Hard- und Software, als auch Funktionsweise zu gewinnen.

Die physikalische Verbindung über ein Netzwerk bietet nicht nur die Möglichkeit über große Distanzen zu kommunizieren und somit Daten zu übertragen und Steuerungsaufgaben auszuführen, sondern verschafft auch eine erweiterte Form der Benutzerschnittstelle zur Benützung des Gerätes durch die Verbindung mit einem Computer. Über ein GUI können dem Anwender die Messdaten präsentiert, sowie Steueraufgaben durch den Einbau von Bedienelementen der grafischen Benutzeroberfläche übersichtlich und einfach durchgeführt werden.

4.1 OSI und Internet Schichtmodell

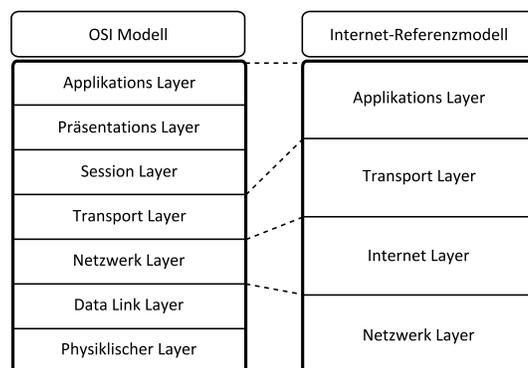


Abbildung 4.1: OSI vs. Internetschichtmodell

Das standardisierte OSI Schichtmodell (vgl. Abb. 4.1) beschreibt den Grundablauf einer Kommunikation in einzelnen Schichten, durch Aufteilung der Kommunikation in Teilaufgaben und bildet somit die Grundlage für Protokolle, die Regeln festlegen, wie Daten nach

dieser Struktur verarbeitet werden.

Es kommt zu einer Schichtung von Protokollen, wobei jede Schicht, angefangen von der Obersten ihre ganz speziellen Informationen hinzufügt. So werden die Daten immer weiter gekapselt und können schließlich über ein physikalisches Medium übertragen werden(vgl.Abb. 4.2). Vorteile einer Schichtung ist eine größere Flexibilität bei Änderungen und Erweiterungen. Der Empfänger muss die ankommenden Daten schichtweise von unten nach oben auswertet und durcharbeiten, um an die eigentliche Nutzinformation zu gelangen.

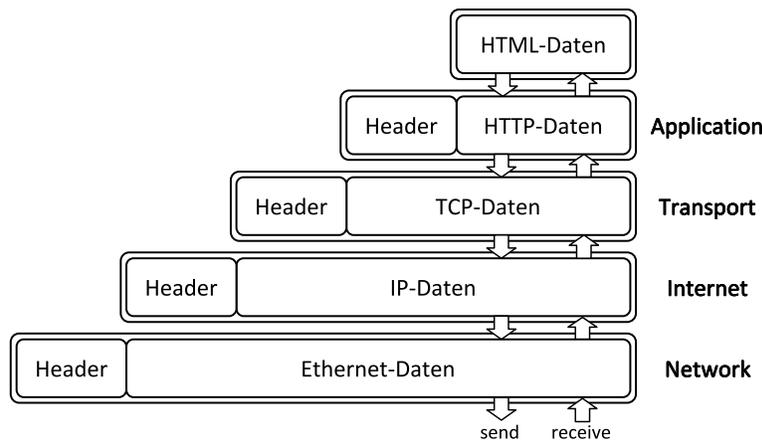


Abbildung 4.2: Datenkapselung

Die Schichtung der Kommunikationsaufgaben nennt man Protokoll Stack. Der Umgang und die Bearbeitung der, durch die Schichtung und Kapselung entstehenden zusätzlichen Informationen, ist die zentrale Aufgabe für die Durchführung der Netzwerkkommunikation. Vergleicht man das OSI-Schichtmodell für allgemeine Kommunikationsaufgaben mit dem Internet Referenzmodell, erkennt man, dass gewisse OSI Schichten zu einer Internet Schicht zusammengefasst werden, aufgrund der zeitlich früheren Entstehung des Internet Modells. Eine Übersicht der Schichten des Internetreferenzmodells, deren Funktionen und der wichtigsten Protokolle bietet Tabelle 4.1.

Layer	Funktion	Protokolle
Application	Von Anwendung definiert	HTTP DHCP
Transport	Kommunikation zwischen Endpunkten	TCP UDP
Internet	Zustellung Routing	IP ARP
Network	Datenübertragung	Ethernet 802.3

Tabelle 4.1: Tabelle Internet-Schichten, Funktionen, Protokolle

4.2 Hardware

Die Hardware einer netzwerkfähigen Embedded Anwendung besteht prinzipiell aus folgenden, in der Abbildung 4.3 ersichtlichen, notwendigen Elementen, die ihrerseits in den unterschiedlichsten Kombinationen zusammengefasst sein können.

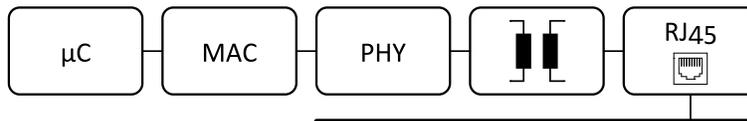


Abbildung 4.3: Netzwerkhardware

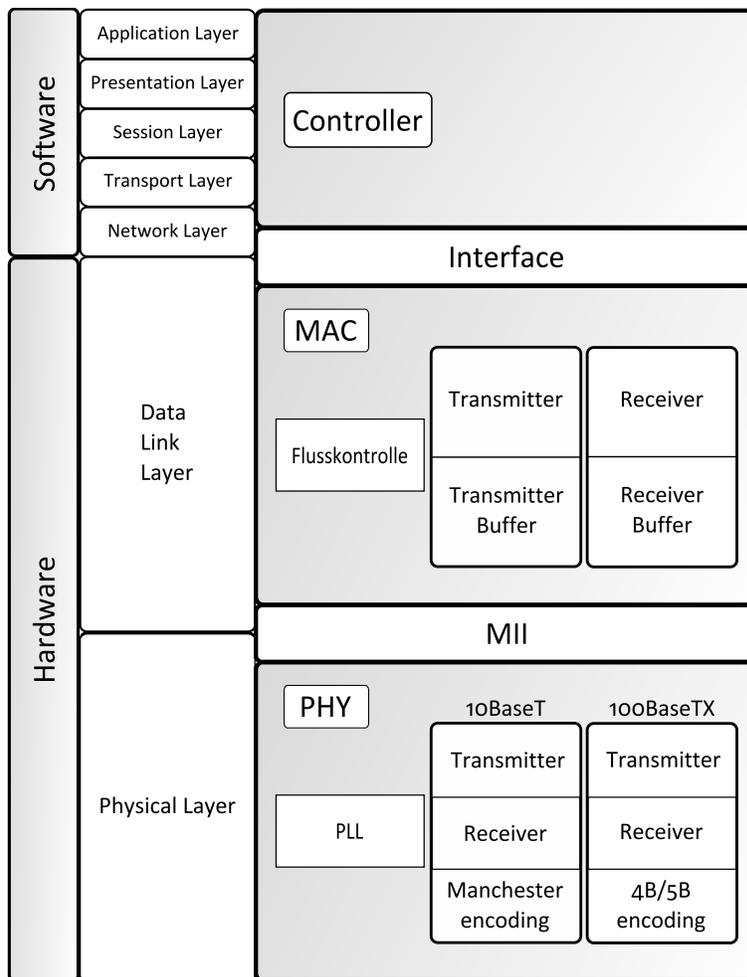


Abbildung 4.4: Elemente und Blöcke bzw. Zuordnung zum Modell

Die galvanische Trennung vom Netz erfolgt über einen Pulstransformator, dessen Übertragungsverhältnis an die Vorgaben und das Spannungsniveau der folgenden Hardware angepasst sein muss. Das physikalische Interface bildet das PHY, das einen Transceiver auf der untersten Schicht des OSI-Modells darstellt. Mit einer Phase-Locked-Loop ist es unter

einer kleinen Phasenabweichung möglich, aus dem ankommenden Signal den Arbeitstakt zu bekommen. Das Datensignal muss dazu aber auch in einem Self-clocking Code codiert sein, was für 10BaseT die Manchester-Codierung und für 100BaseTX der 4B/5B Code ist. Die MII (Media Independent Interface) ist die Schnittstelle des PHY mit dem Media Access Controller MAC, der die nächsthöhere Ebene des OSI-Modells, die Data Link Schicht repräsentiert. Der MAC regelt den Zugriff auf den Übertragungskanal, speichert zu sendende und empfangene Daten zwischen und sorgt mit der Flusskontrolle für eine dynamische Steuerung des Sendeverhaltens der Gegenseite. Er verfügt über eine feste, im Netzwerk einzigartige Adresse, die sogenannte MAC Adresse, mit der er sich über das Netzwerk eindeutig identifizieren lässt. Ein Mikrocontroller implementiert über Software die weitere Datenverarbeitung der nachfolgenden Schichten des Protokollstapels gemäß des OSI-Modells, bis zur eigentlichen Anwendung auf oberster Ebene. Eine Übersicht über die Komponenten, deren Aufgaben und Zuordnung zum OSI-Modell bietet Abbildung 4.4.

Die einzelnen Funktionselemente für eine Netzwerkkommunikation finden sich auch in unterschiedlichen Zusammenfassungen in einzelnen Bauteilen wieder. So sind beispielsweise MAC und Mikrocontroller zu sogenannten Single-Chip Lösungen zusammengefasst, oder PHY und Mac zu einem Ethernet Transceiver oder Ethernet Controller, oder es existieren Komplettlösungen, die vom RJ45 Stecker bis zum Controller zur Bearbeitung des Protokoll Stapels inklusive Software alles ineinander vereinen. Diese Komplettlösungen bieten dann zusätzlich zur Ethernet-Schnittstelle ein weiteres Interface zur Anbindung einer Embedded Applikation.

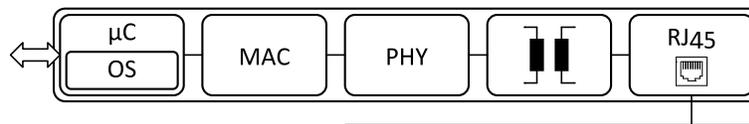


Abbildung 4.5: Komplettlösung Netzwerkhardware mit Schnittstelle

Beide Ansätze bieten Vor- und Nachteile. So ermöglicht der Aufbau mit einzelnen Komponenten im Vergleich zur Komplettlösung größere Flexibilität zur eigentlichen Applikation durch Anpassung von Soft- und Hardware, bietet aber eventuell einen kleineren Funktionsumfang des IP Stacks, aufgrund der gegebenen Ressourcen. Komplettlösungen sind im Allgemeinen teurer, platzsparend und durch die Integration aller nötiger Komponenten einfach in bestehende Applikationen zu implementieren. Die Entscheidung hängt aber hauptsächlich von der Art der gewünschten Webfunktionalität und den damit verbundenen Anforderungen ab.

4.3 Ethernet

Allgemein nimmt die Netzwerkfähigkeit und Kompatibilität von elektronischen Geräten durch hardwareunabhängige Standards und Protokolle aus mehreren Gründen zum generellen Vorteil zu. Ethernet ist in diesem Zusammenhang zu einer Art Standard Schnittstelle für Lokale Netzwerke geworden. Durch Schnittstellenadapter lassen sich auch verschiedene Kommunikationsnetze und Busse an ein solches Netzwerk anschließen, was zu einer Vielzahl an unterschiedlichen Teilnehmern führt. Nach dem OSI-Modell umfasst das Ethernet sowohl die unterste physikalische Schicht als auch die darüber liegende Data Link Schicht,

da der Begriff Ethernet sowohl die Art der Bitübertragung, als auch die Pakete und Protokollformen bezeichnet. Im Internet Referenzmodell entspricht das der Netzwerkschicht.

Die physikalische Datenübertragung, die Form und die Medien, die dazu verwendet werden sind nach IEEE 802.3 standardisiert. Da alle Teilnehmer sich den gleichen Kanal teilen, ist ein bestimmtes Zugriffsverfahren für die Übertragung notwendig. Mithilfe des Zugriffs über das statistische CSMA/CD (Carrier Sense Multiple Access with Collision Detection) Verfahren hört jede Station am Übertragungsmedium mit, ob dieses frei ist, und belegt es, wenn eine Übertragungsabsicht besteht. Dieser Prozess ist nicht deterministisch, also lässt sich keine Aussage über die tatsächliche Übertragungsdauer treffen. Nach einem bestimmten Algorithmus versucht nun die Station, falls der Kanal belegt war, in wechselnden Abständen den Sendewunsch nachzukommen. Falls kein Sendevorgang zustande kommt, wird die nächsthöhere Protokollschicht davon unterrichtet. Aufgrund der Ausbreitungsverzögerung im Medium kann es zum gleichzeitigen Senden mehrerer Teilnehmer und damit zu Kollisionen kommen, die von den Teilnehmern als solche registriert und die Sendevorgänge eingestellt werden. Nach einer zufälligen Zeitspanne wird von den Teilnehmern wieder ein Sendevorgang versucht, falls der Kanal inzwischen nicht wieder belegt ist. Mit steigender Anzahl der Netzwerkteilnehmer nimmt aber zwangsläufig die Anzahl der Kollisionen zu. Abhilfe schafft eine Aufteilung der Netze durch Switches in geschaltete Teilnetze.

Nach der genau festgesetzten 7 Byte langen Präambel und dem Start Frame Delimiter, die den Unterschied zwischen Frame und Packet ausmachen, folgt die Empfänger- und Sender-Adresse. Über die 6 Byte lange eindeutige MAC Adresse lassen sich die Daten-Frames dem Empfänger und dem Sender zuordnen. Die Mindestlänge eines Frames beträgt 64 Byte und wird gegebenenfalls durch Füll-Bytes erreicht. Die Maximallänge ist auf 1518 Byte beschränkt.

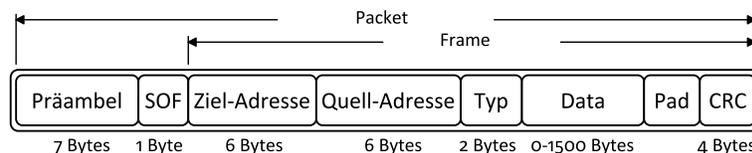


Abbildung 4.6: Ethernetframe

Die folgenden zwei Bytes geben den Typ und somit das Protokoll des vorliegenden Frames an, gefolgt von den Nutzdaten mit eventuellen Füll-Bytes und der abschließenden 4 Byte langen Checksumme, um den Frame und seine Daten auf Fehler zu überprüfen. Die Pakete können über die Empfängeradresse an einen einzigen Teilnehmer versendet werden, Unicast genannt, oder an alle angeschlossenen Teilnehmer über einen Broadcast. Der designierte Empfänger verfügt über einen Adressfilter, der die Datenpakete nur im Falle einer Übereinstimmung mit der Empfängeradresse oder der Broadcast Adresse durchlässt.

4.4 Netzwerkprotokolle

Die Kommunikation im Netzwerk folgt gewissen Regeln und Vorschriften, die durch standardisierte Protokolle nach den verschiedenen Ebenen bzw. Schichten des Kommunikationsmodells organisiert sind. Es existieren mehrere verschiedene Protokolle für unterschied-

liche Zwecke und Aufgaben, auch auf gleicher Modellebene. Das Zusammenspiel mehrerer Protokolle aller Ebenen und deren Datenkapselung stellen den gesamten Ablauf einer Kommunikation dar. Zusammengefasst folgt eine kurze Übersicht über wichtige Protokolle, die in dieser Arbeit Anwendung finden.

IP Die Netzwerkteilnehmer haben neben der eindeutigen MAC Adresse eine änderbare logische IP Adresse zur Strukturierung des Gesamtnetzwerks, was die Voraussetzung für Routing, die intelligente Weiterleitung von Paketen ist. Diese 32 Bit lange Adresse bildet die Quell- und Zielpunkte des IP Protokolls (Internet Protocol), das auf der Ebene der Internet Schicht, dem Austausch der IP-Datagramme und deren Adressierung in unterschiedlichen Netzwerken dient. Dabei gibt es auf dieser Ebene keine Mechanismen zur korrekten und verlässlichen Übertragung oder für den Empfang der Pakete in der richtigen Reihenfolge, weshalb die Zuverlässigkeit von den darüber liegenden Protokollen der höheren Schicht gewährt werden muss. Die maximale Paketgröße beträgt dabei 64 KByte, wobei üblicherweise die Größe auf die maximale Datenlänge des gebrauchten Kanals über die MTU (Maximum Transfer Unit) begrenzt wird, um Fragmentierungen zu vermeiden. Im Fall der Übertragung über Ethernet, ist man an dessen maximale Nutzdatenlänge von 1500 Byte gebunden. Neben den IP Adressen von Quell- und Ziel-Host umfasst die Protokollstruktur auch ein Feld, das die Art des Transport Protokolls bestimmt, zur Weiterleitung des Pakets an diese höhere Protokollschicht.

ARP Ein weiteres Protokoll der Internet Schicht ist das ARP Protokoll (Address Resolution Protocol), dessen Ziel es ist, die für die Kommunikation nötige eindeutige MAC Adresse auf Grund der bekannten IP Adresse des Hosts herauszufinden, da für die eigentliche Datenübertragung über Ethernet die physikalische MAC Adresse notwendig ist. Die geschieht über einen Anfrage-Antwort-Prozess.

TCP Das TCP Protokoll (Transmission Control Protocol) ist nach dem Kommunikationsmodell auf der Ebene der Transportschicht angesiedelt und sorgt für den eigentlichen paketweisen Datentransport zwischen zwei Endpunkten. Es ergänzt die übrigen Protokolle der anderen Schichten um den Mechanismus der Datensicherheit und Datenflusskontrolle. Damit ist es gerüstet gegen verloren gegangene, fehlerhafte und in falscher Reihenfolge gelieferte Pakete und verantwortlich für den Auf- und Abbau einer Verbindung sowie dem Handshaking zwischen den Client, der die Verbindung öffnen will und dem Server, der auf eine Anfrage wartet. Die Daten der darüberliegenden Anwenderschicht werden dazu in einzelne Segmente zerteilt, die in weiterer Folge auf der darunterliegenden Schicht als IP-Datagramme bezeichnet werden, woraus der Zusammenhang mit dem IP Protokoll ersichtlich ist. Oft wird deshalb von TCP/IP gesprochen. Die Segmente werden mit Nummern bezeichnet, wodurch sie beim Empfänger wieder richtig zusammengesetzt und auf Vollständigkeit überprüft werden. Den Erhalt des bestimmten Paketes muss der Empfänger dann quittieren. Ohne diese Bestätigung des Empfängers, ist der Sender nach einer gewissen Zeit wieder veranlasst, die Übertragung zu wiederholen.

Zur Kommunikation wird die Verbindung im eigentlichen Sinne erst nach der Durchführung eines bestimmten Musters, dem Drei-Wege-Handshake aufgebaut. Nach dem Aufbau kann eigentlich nicht mehr von Client und Server gesprochen werden, da die

Verbindung für beide Hosts gleichwertig ist. Die Datenübertragung erfolgt dabei solange, bis einer der Teilnehmer die Verbindung nach demselben Schema wieder beendet. Zusätzlich zu den IP Adressen verwendet das TCP Protokoll noch Quell- und Ziel Ports, über die die Daten an die darüberliegende Anwendungsschicht weitergeleitet und so unterschiedliche Datenströme voneinander getrennt werden. In diesem Zusammenhang existieren die sogenannten Well-known Ports, die für bestimmte Anwendungen reserviert sind. Die Kombination einer IP Adresse und eines Ports stellt somit einen klar definierten Kommunikationsendpunkt dar, über welchen Anwendungen Daten austauschen, und wird als Socket bezeichnet.

UDP Das User Datagram Protocol ist ein minimales und im Gegensatz zum TCP Protokoll ein nicht zuverlässiges Protokoll mit wenig Overhead und somit zur schnellen Datenübertragung geeignet. Zur Weiterleitung an die obere Anwenderschicht werden wie beim TCP Protokoll Ports verwendet.

DHCP Mit dem Dynamic Host Configuration Protocol auf der Ebene der Anwendungsschicht, wird einem neuen Netzwerkteilnehmer durch einen DHCP-Server, wie beispielsweise einen Router, automatisch die Initialisierungskonfiguration mit der Zuweisung einer zeitlich begrenzten IP Adresse überwiesen. Dabei ist es auch möglich, dem Client mittels der MAC Adresse immer eine gleiche IP Adresse zuzuteilen. Weiters werden noch die Netzmaske, die mit der IP Adresse verknüpft wird und diese, zur Unterscheidung interner und externer Netzteilnehmer in einen Netzwerkteil und Geräteteil trennt, sowie die Gateway Adresse und die Adresse des DNS-Servers vergeben. Gateways sind dabei vergleichbar mit Routern, wobei Gateways durch Protokollumsetzung auch Netze mit unterschiedlichen physikalischen Eigenschaften verbinden können und somit ein Knotenpunkt für externe Teilnehmer sind. Der DNS (Domain Name System) Server wandelt den Domainnamen in die zugehörige IP Adresse um (forward lookup).

HTTP Das Hypertext Transfer Protocol ist auf der Ebene der Anwendungsschicht angesiedelt und wird vom Browser, der das Anwendungsprogramm auf höchster Ebene bildet, benutzt. Dabei dient es dem Datentransfer über das Netz. Bei den Daten handelt es sich hauptsächlich um HTML (Hyper Text Markup Language) Seiten und darauf eingebundene Inhalte, wie beispielsweise Programme in Java. Zuverlässigkeit ist dabei erst durch ein Protokoll der Transportschicht wie TCP gegeben. Das Prinzip des Datenaustausches erfolgt dabei nach einem Request and Response Schema nach gewissen Methoden. Für Anfragen des Clients nach einem bestimmten Inhalt oder einer Datei auf einem Server, spezifiziert durch eine URL, ist dies meist die GET-Methode.

4.5 TCP/IP Stack

Den Aufbau und die Abarbeitung eines Protokollstapels, mit den verwendeten Protokollen der einzelnen Ebenen, müssen beide Hosts für die Durchführung der Kommunikation beherrschen. Bei Mikrocontrollersystemen wird dabei auf Overhead und Mechanismen, die für die gewünschte Art der Verbindung nicht unbedingt erforderlich sind, verzichtet. Trotzdem müssen auch sie die Erzeugung und den Umgang der notwendigen Protokolle

und deren Spezifikation verstehen. Von dieser Implementierung hängen die Zuverlässigkeit und die Möglichkeiten der gesamten Kommunikation ab.

TCP/IP Stacks sind in den verschiedensten Varianten, mit unterschiedlichem Umfang, kommerziell oder kostenlos, für den Einsatz in diversen Hardware Plattformen verfügbar. Programmspeicher für Webseiten und Stack, RAM für die Zwischenspeicherung von Daten-Frames und Rechenleistung für deren Abarbeitung und Erstellung, sind dabei die Hauptkriterien, die an die Hardware gestellt werden. Mit „ μ IP“ und „lwIP“ seien hier beispielhaft zwei Vertreter von TCP/IP Stacks genannt, die kommentiert, portabel und frei verfügbar sind, sowie geringe Anforderungen stellen. Als Ethernetcontroller im Zusammenhang mit einer diskreten Lösung, also den Aufbau mit einzelnen nötigen Bauteilen oder Gruppen zur Netzwerkkommunikation, empfehlen sich die ebenfalls gebräuchlichen Modelle ENC28J60 oder CS8900A. Bevorzugt man eine Komplettlösung, ist an erster Stelle das XPort der Firma Lantronix zu nennen.

4.6 XPort

Über die Ethernet-Schnittstelle sollen die mit der Messschaltung aufgenommenen und im Mikrocontroller verarbeiteten Messdaten im Netzwerk bzw. Internet übertragen, als auch gewisse Funktionen des Messgerätes aus der Ferne gesteuert werden. Dazu bedarf es eines Embedded Webservers, der mit den, für die Kommunikation im Netzwerk nötigen Protokollen, umzugehen versteht. Am anderen Ende der Kommunikationskette soll jeder beliebige Rechner im Netzwerk mittels eines Browsers, ohne zusätzlicher lokal abgespeicherter Software darauf Zugriff haben und selbst Daten über den Server an den Mikrocontroller versenden können.

Ermöglicht wird dies durch ein Java Applet, das auf dem Webserver, eingebettet in eine Webseite, abgelegt und nach dem Laden durch den Browser, direkt in diesem ausgeführt wird. Dabei öffnet es über ein Socket einen Kanal zum Server, der dann einen Knotenpunkt für den Datenaustausch zwischen dem Java Applet Programm und dem Mikrocontroller verkörpert. Für diese Aufgabe bietet sich das XPort von der Firma Lantronix als kompakte, zuverlässige und vielseitige Lösung an.

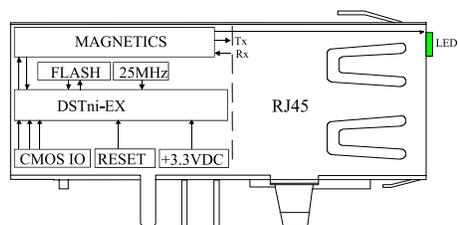


Abbildung 4.7: Darstellung und Aufbau XPort[Lan05].

Das XPort ist ein kleiner, hoch integrierter, kompletter Server, der in einer RJ45 Buchse Platz findet, und stellt eine Komplettlösung für den Netzwerkzugang da. Es verfügt neben einem 10/100 Ethernet Transceiver, der auf der Ebene der Netzwerkschicht arbeitet, auch über die nötige Hard- und Software zur Abarbeitung der Protokolle der oberen Schichten sowie über eine serielle Schnittstelle und Status und Diagnose LEDs, sodass es keiner zusätzlichen externen Hardware bedarf. Das XPort bildet quasi eine Brücke zwischen einer

seriellen Schnittstelle (RS232) und einer Ethernet-Schnittstelle, durch die einfache bidirektionale Weiterleitung der eingehenden Daten zwischen dem Port 10001 und der seriellen Schnittstelle. Die seriell empfangenen Daten werden dazu einfach in ein TCP oder UDP Packet eingekapselt und so über ein IP basierendes Netzwerk verschickt.

Zusätzlich stellt Lantronix eine Software bereit, die ein virtuelles COM Port am PC erzeugt und somit die Illusion einer reinen seriellen Verbindung vortäuscht. In diesem Zusammenhang kann auch von COM-LAN Adapter gesprochen werden. Da das XPort auch vermag aktiv Verbindungen herstellen, bilden zwei über das Netzwerk verbundene COM-LAN Adapter eine vollständige Datenbrücke für eine serielle Verbindung. Da man davon ausgehen kann, dass die mögliche Übertragungsgeschwindigkeit im Netzwerk größer als die der seriellen Schnittstelle zwischen XPort und Mikrocontroller ist, wird der maximal mögliche Durchsatz der Verbindung dadurch begrenzt.

Um die Messdaten und das Interface mit dem XPort, für den Anwender in Form einer ansehnlichen und interaktiven Oberfläche darzustellen, und von lokaler, am Rechner installierter Software unabhängig zu sein, wird das XPort als Webserver betrieben, der dem Client eine Webseite und ein darauf eingebettetes Java Applet, das auf die Ressourcen des XPort zugreifen kann, anbietet. Dabei werden mehrere Ports verwendet, die Protokolle an die oberen Schichten adressieren und einen Kommunikationsendpunkt darstellen. Der nötige TCP/IP Stack und ein Betriebssystem zur Steuerung sind bereits integriert, sodass das XPort eine Komplettlösung aus Hardware und Software darstellt.

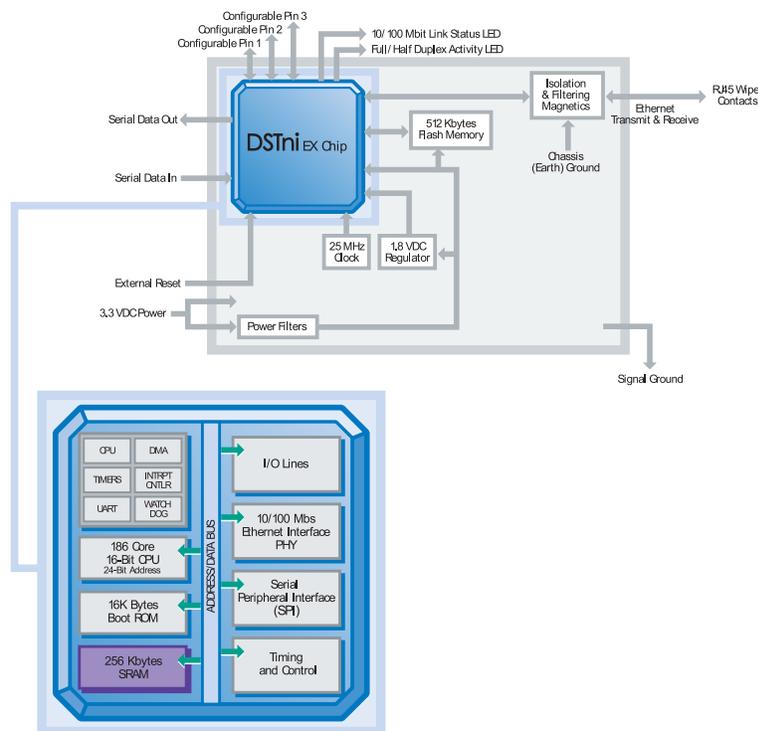


Abbildung 4.8: Blockschaltbild XPort[Lan08].

Die CPU ist ein DSTni-EX Controller der über 256 KB SRAM und 512 KB an Flash Speicher verfügt (vgl. Abb. 4.8). Dem Server stehen davon 384 KB, unterteilt in sechs 64

KB große Segmente, zur Speicherung der Webseiten zur Verfügung. Mit dieser Aufteilung des Speichers ergibt sich eine maximal mögliche Dateigröße von 64 KB pro Webseite. Dieses Kriterium ist in Hinblick auf die HTML Seite und die Dateigröße der einzelnen Files des darauf eingegliederten Java Applet zu beachten. Eine der Zwei in die Buchse integrierten Bi-Color LEDs, die Link LED, zeigt die Datenrate durch ihre Farbe an. Leuchtet sie grün, beträgt die Datenrate des Netzwerks 100Mbps (Fast Ethernet), bei orangen Licht 10Mbps (Ethernet). Diese Ethernetvarianten unterscheiden sich nicht nur durch ihre Übertragungsrates sondern auch durch ihre Leitungscodierung. Die andere LED, die Activity LED, gibt anhand ihrer Farbe Auskunft über die Art der Verbindung, wobei Grün für den üblichen Vollduplexbetrieb, dem gleichzeitigen Senden und Empfangen steht und Orange für den Halbduplexbetrieb, dem alternierenden Senden und Empfangen von Daten. Das Blinken dieser LED zeigt die momentane Aktivität einer Verbindung an. Bei der XE-Variante des XPort kann diese nur unverschlüsselt erfolgen, wohingegen das SE-Derivat, mittels des SSL Verschlüsselungsprotokoll, das auf der Transportschicht des Internetmodells anzusiedeln ist, eine sichere Verbindung bietet.

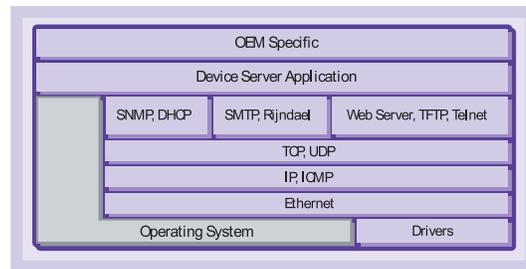


Abbildung 4.9: Softwareschichtenmodell XPort[Lan08].

Die Struktur der Software des XPort entspricht dem abgebildeten Schichtenmodell (Abb. 4.9). Die zwei obersten Schichten sind anwenderspezifisch programmierbar und bilden das eigentliche Anwendungsprogramm, das die nötigen Funktionen bereitstellt und die gewünschten Aufgaben erledigt. Das integrierte Betriebssystem des XPort führt die, vom Anwender verborgene Abarbeitung der Protokolle der einzelnen Schichten aus, welche zusammen den Protokollstapel nach dem OSI Referenzmodells bilden. Das IP Protokoll wird verwendet um die Verbindung der Teilnehmer zu ermöglichen und bildet mit dem um eine Schicht höher gelegenen TCP Protokoll, das die Richtigkeit und Ordnung der Daten gewährleistet die Basis der Netzwerkkommunikation. Zusätzlich wird das DHCP Protokoll (Anwendungsschicht des Internetschichtenmodells) vom XPort benutzt, um Daten mit dem DHCP Server mittels UDP Protokoll (Transportschicht) auszutauschen. Dabei handelt es sich um Informationen, wie die durch den DHCP Server zugeteilte IP Adresse, Subnet Maske, Gateway Adresse und die Adresse des DNS Servers.

Mittels des HTTP Protokoll stellt der Server nach GET Anfragen des Clientbrowsers über den Port 80 dem Client seine Inhalte bereit. Durch den Aufruf der HTML Seite wird auch das Java Applet heruntergeladen, das den weiteren Datenfluss über ein Socket zwischen den nunmehrigen Kommunikationsendpunkten Applet und Mikrocontroller von sich aus steuert. Weiters können von der Ebene der Anwendungsschicht, das Telnet Protokoll zu Konfigurationszwecken, das SMTP Protokoll zum Austausch von E-mails und das TFTP Protokoll zum Speichern von Firmware und Webseiten im Flash genutzt werden.

Die Fähigkeiten des XPort umfassen auch noch die Kenntnis des SNMP, ICMP und ARP Protokolls.

XPort Features:

- Complete, integrated solution in an RJ45 form factor
- Embedded web server
- 10/100Mbit Ethernet - auto-sensing
- Stable, field proven TCP/IP protocol suite and web-based application framework
- Easy configuration through a web interface
- Easy customization of HTML web pages and configuration screens
- E-mail
- 128, 192, or 256-bit AES Rijndael encryption (optional)
- EMI tested and compliant
- Extended operating temperature: -40 to +85 C normal mode -40 to +75 C high-performance mode
- High-performance processor (12 MIPS at 48 MHz, 22 MIPS at 88 MHz)
- Network overhead handled by XPort
- Password protection
- Upgrade XPort's firmware over the network
- 3.3V power
- Serial-to-10/100 Ethernet conversion
- 921,600 baud serial speed

Tabelle 4.2: Features XPort[Lan05].

4.6.1 Konfiguration

Damit das XPort im Netzwerk korrekt funktioniert, muss es eine im Netzwerk einmalige IP Adresse haben. Standardmäßig bezieht das XPort diese über das DHCP Protokoll von einem DHCP Server im Netzwerk. Hierbei handelt es sich aber nur um eine geleaste IP, die somit bei jeder Ankopplung an das Netzwerk verschieden sein kann. Um aber vom Netzwerk aus immer auf das XPort, beziehungsweise auf dessen Webserver zugreifen zu können, muss dessen IP bekannt sein, wie durch eine explizite Vergabe innerhalb des lokalen Netzwerkes erreicht wird. Es gibt prinzipiell folgende Möglichkeiten das XPort zu konfigurieren und an eine vorliegende Umgebung anzupassen:

DeviceInstaller ist ein von Lantronix zur Verfügung gestelltes Programm, das alle an das Netzwerk angeschlossenen XPorts findet und deren momentane Einstellungen, wie auch IP Adressen anzeigt bzw. die Zuweisung solcher durch den Anwender unterstützt.

WebManager ist ein browserbasierendes Konfigurationswerkzeug von Lantronix und standardmäßig auf den ausgelieferten XPort im Speicherbereich für Webseiten abgelegt. Es wird im Browser nach dem Aufruf des XPort geladen und bietet alle Möglichkeiten der Konfiguration auf der Netzwerkseite, wie IP, Subnetmask, Gateway, Hosts und Verbindungsoptionen sowie auf der Seite der seriellen Schnittstelle. Bei Verwendung des XPort als Webserver werden allerdings die anwenderspezifischen Inhalte im gleichen Speicherraum abgelegt, sodass der darin enthaltene WebManager überschrieben wird.

Über eine *Telnet Verbindung* besteht die Möglichkeiten Konfigurationseinstellungen über das Netzwerk vorzunehmen. Dazu öffnet man über die Kommandozeile mit folgender Eingabe eine Verbindung an Port 9999, wobei x.x.x.x für die IP Adresse des XPort steht:

```
telnet x.x.x.x 9999
```

Für eine Telnet Verbindung ist bereits die Kenntnis der IP Adresse des XPort nötig. Der DeviceInstaller, der die IP Adressen des XPort kennt, bietet daher die Möglichkeit diese Verbindung direkt aufzurufen.

Die *serielle Schnittstelle* des XPort bietet eine weitere Möglichkeiten um lokal Konfigurationseinstellungen vorzunehmen und kann individuell konfiguriert und so an die Prozessumgebung angepasst werden. Standardmäßig sind Baudraten zwischen 300 und 230400bps möglich, wobei durch eine Übertaktung der CPU auch 460Kbps und 920Kbps erreichbar sind. Dies kann einfach im Konfigurationsmenü durchgeführt werden, woraufhin nicht nur der CPU Takt auf 88MHz ansteigt, sondern auch die Stromaufnahme und die produzierte Abwärme. Des Weiteren stehen auch noch Einstellungsmöglichkeiten, wie Anzahl der Daten- und Stopbits, die Auswahl eines Paritätsbits und der Flusskontrolle zur Verfügung. Damit ist das XPort in der Lage an die serielle Schnittstelle jedes beliebigen Gerätes, angepasst zu werden und diesem eine Anschlussmöglichkeit an das Ethernet zu eröffnen.

Zusätzlich verfügt das XPort auch über drei frei konfigurierbare *GPIO Pins*, die entweder in Verbindung mit der UART für eine Art Handshaking verwendet werden können, oder völlig unabhängig davon als Ein- oder Ausgabe-Pins durch die Anwendersoftware kontrolliert werden. Dies erfolgt mittels Kommandos nach einem fest definierten Protokoll über eine Socket-Verbindung an Port 30704.

4.6.2 Implementierung

Da das XPort von seinem Aufbau her quasi als eigenständiges System gesehen werden kann, ist die Integration selbst bei bestehender Schaltung einfach, da so gut wie keine externe Beschaltung von Nöten ist. Die Betriebsspannung des XPort beträgt 3.3 V, ein interner Spannungsregler versorgt den Controllerkern mit 1.8 V. Der Versorgungsstrom im idle Zustand und bei normalem Betrieb des Kerns mit 48 MHz Taktfrequenz beträgt laut Datenblatt 119 mA. Bei Netzwerkzugriffen steigt der Strombedarf je nach Übertragungsgeschwindigkeit und Art der Ethernetverbindung auf 224 bzw. 190 mA bei 10Base-T und 100Base-T an. Fällt die Versorgungsspannung unter 2.7 V ab, wird ein interner Reset durch eine Überwachungsschaltung ausgelöst. Der externe Reset Pin, dessen aktiver Pegel

low ist, kann über einen Pull-up Widerstand an die 3.3 V Versorgung des Bausteines gelegt werden, oder an einen Pin des Mikrocontrollers zur externen Reset-Steuerung. Trotz der 3.3 V Pegel an den Pins sind alle Signal-Portpins tolerant gegenüber 5 V, sodass das XPort direkt an die serielle Schnittstelle des Mikrocontrollers angeschlossen werden kann.

Die Chassis Pins oder Schild Pins haben zwei Funktionen. Zum einen dienen sie als wichtige Wärmesenke und sollten deshalb laut Datenblatt zur Wärmeableitung mit einer 1sq Inch großen Kupferfläche der Leiterplatte verbunden werden. Weiters ist das Chassis, als exponiertes Teil, aufgrund des Ethernet-Steckers, möglichen ESD Vorfällen ausgesetzt. Zur Vermeidung von Beschädigungen kann das Chassis durch einen Kondensator, mit einem geringen äquivalenten Serienwiderstand und einer hohen Spannungsfestigkeit, sowie einer Zenerdiode oder einer Suppressordiode gegen Erde verbunden werden.

4.6.3 Speicherung der Webseiten

Nach dem Erstellen einer HTML Webseite, welche das Java Applet inkludiert, werden die *.html und die *.class Dateien des Applets mit der von Lantronix bereitgestellten Applikation „web2cob“ in *.cob Dateien umgewandelt, die nicht größer sein dürfen, als die einzelnen Speicherbänke des Webservers. Die Übergabeparameter bei Programmaufruf sind das Quellverzeichnis und die Ausgabedatei:

```
web2cob /d <directory> /o <output*.cob>
```

Das Programm versucht dabei alle in einem Ordner abgelegten Daten zu einer COB-Datei zu binden. Die *.class Dateien des Java Applets können dabei, falls durch die Dateigröße nötig, auf mehrere COB-Dateien aufgeteilt werden. Die entstandenen Dateien werden dann mittels eines tftp Client Programms ins Flash des XPort übertragen und zwar jeweils in eine der sechs, je 64KB großen Speicherbänke. Diese werden mit WEB1 bis WEB6 bezeichnet. Über die Kommandozeile kann dies mit folgender Eingabe durchgeführt werden:

```
tftp -i <ip address> put <local filename> <destination file name>
```

Im Anschluss kann die am Server deponierte HTML-Seite über den Browser, mit der Eingabe der IP Adresse des Webservers und dem Namen, unter dem die HTML-Seite abgespeichert wurde, aufgerufen werden.

4.6.4 Web Server Grundlagen

Die Kommunikation zwischen Client und Server verläuft nach einem Anforderung-Antwort-Prinzip. Der HTTP Webserver des XPort unterstützt GET- und HEAD-Methoden des Clients. Dabei handelt es sich um die zwei Standard HTTP Request Verfahren des Clients. Ein Request nach einem bestimmten Inhalt, der durch eine URL (Uniform Resource Identifier) spezifiziert ist, erfolgt mittels eines Browsers. Die GET-Methode fordert dabei die nachgefragte Datei vom Server an. Die HEAD-Methode hingegen verlangt nur nach den Header Informationen dieser Datei, womit durch den Client geprüft werden kann, ob eine bereits zwischengespeicherte Kopie dieser Datei sich mittlerweile verändert hat. Eine weitere Form des Requests, die POST-Methode, die der Übertragung von Informationen vom Client zum Webserver dient und diese mittels CGI (Common Gateway Interface) an

eine weitere Server-seitige Applikation zur Bearbeitung weitergibt, wird durch das XPort nicht unterstützt. Der HTTP Server des XPort gibt lediglich statische Dokumente oder Daten an den anfordernden Browser weiter. Um kontinuierlich anfallende Daten, vom Ort an dem sie entstehen, in Echtzeit über das Netz zu übertragen und auf der Clientseite darzustellen, bieten sich neben der Möglichkeit eines anwenderspezifischen Programms, das am Clientcomputer zur Verfügung steht und sich über das Netzwerk mit dem Server verbindet, der daraufhin eine Bypass Funktion für Daten darstellt, noch die des Java Applets.

Mit der objektorientierten Programmiersprache Java und der Programmform, der Java Applets, ist es möglich, Programme auf Webseiten einzubetten und direkt im Browser auszuführen. Voraussetzung ist, dass dieser über ein Java-Plug-in verfügt, das eine virtuelle Maschine bereitstellt, die die Laufzeitumgebung für die Programme ist. Da eine virtuelle Maschine für viele verschiedene Plattformen existiert, ergibt sich sozusagen ein systemunabhängiger Charakter der Programme. Die verfügbaren und umfangreichen Bibliotheken, auch hinsichtlich Webanwendungen und grafischen Gestaltungsmöglichkeiten von Benutzeroberflächen, implementieren eine große Bandbreite von Methoden zur Codegenerierung. Ein Java Applet erzeugt dabei zur Kommunikation mit dem Server sein eigenes Socket, über das auf der Ebene der Anwendungsschicht protokollunabhängig Daten ausgetauscht werden können. Der offensichtliche Vorteil dieser Variante ist, dass das Programm nicht bereits am Client lokal vorhanden sein muss, sondern auf einer HTML Seite eingebettet ist und beim Laden dieser Seite mitgeladen wird. Somit kann von jedem Computer aus, der über eine Netzwerkverbindung verfügt, direkt auf den Server und das Gerät zugegriffen werden, wobei das XPort nur einen simultanen Zugriff unterstützt. Die direkte Kommunikation mit dem Mikrocontroller wird dabei erst durch das Java Programm erreicht, das nach dem Herunterladen, über eine offene Socketverbindung zum XPort, dank dessen Eigenschaften der Datenweiterleitung, in der Lage ist auf die serielle Schnittstelle zuzugreifen.

4.6.5 Netzwerkumgebung und Serverzugriff

Standardmäßig verwendet das XPort das DHCP Protokoll um sich vom DHCP Server eine dynamische IP-Adresse zu holen. Die somit zugeteilte IP lässt sich einfach mit dem von Lantronix zur Verfügung gestellten Programm DeviceInstaller ermitteln, das mit einem UDP-Broadcast nach im Netzwerk verfügbaren XPorts sucht. Zu beachten sind allerdings die Firewallinstellungen des Computers, die eine mögliche Erkennung des XPort behindern können. Das Programm DeviceInstaller zeigt zusätzlich zu den im LAN verfügbaren XPorts auch deren momentane Konfiguration an bzw. erlaubt es dem XPort explizit eine IP zuzuweisen.

Eine andere Möglichkeit um ein angeschlossenes XPort am Netzwerk zu identifizieren, wäre mittels eines Broadcast Pings, wobei an alle möglichen Adressen ein Ping gesendet wird, um so alle am Subnetzwerk angeschlossenen Komponenten herauszufinden. Erst wird dieser Broadcast Ping ohne das an das Netzwerk angeschlossene XPort ausgeführt und anschließend mit, woraus aus der Differenzmenge der Adressen die Gesuchte folgt. Das Ping Programm unter Windows unterstützt diese Form des Pings aber nicht.

Die, dem XPort über DHCP zugewiesene IP ist dynamisch und kann sich somit bei jeder Verbindung ändern. Da das XPort aber als Server arbeitet, sollte es immer unter der

gleichen Adresse verfügbar sein. Dazu vergibt man dem XPort selbst eine statische, lokale IP innerhalb des Netzwerks oder konfiguriert den DHCP-Server, beispielsweise Router so, dass er für die bestimmte und einmalige MAC Adresse des XPort immer die gleiche IP vergibt. Die sechs Byte lange MAC Adresse des XPort steht auf dessen Gehäuse und beginnt bei Geräten der Firma Lantronix mit den drei Bytes 00:20:4A, fortgesetzt durch eine einmalige und eindeutige Kombination von drei weiteren Bytes.

Innerhalb eines lokalen Netzwerks sind die am Webserver des XPort abgelegten Inhalte und Webseiten immer unter der lokalen IP Adresse mittels eines Browsers unter Angabe einer URL, die das zu verwendende Protokoll, die Host Adresse und den Pfad beinhaltet, zu lokalisieren und zu erreichen. Auf die Eingabe eines am Server nicht existierenden Pfades folgt die Ausgabe eines „Error 400“, der anzeigt, dass die Anfrage aufgrund ungültiger Syntax vom Server nicht verstanden wurde. Zugriffe mehrerer Teilnehmer zur gleichen Zeit werden hardwaremäßig nicht unterstützt, es erscheint eine Fehlermeldung. Sollte ein Teilnehmer eine offene TCP/IP Session mit dem XPort unterhalten und diese nicht durch Schließen der Webseite und damit des Java Applets gültig beenden oder die Verbindung unterbrechen, ist der Server selbst von dem Verbindungsende nicht in Kenntnis und es dauert erst eine gewisse Zeitspanne, bis er wieder einem anderen Teilnehmer exklusiv zur Verfügung stehen kann.

Für Zugriffe außerhalb des lokalen Netzwerkes unter der öffentlichen IP Adresse, wie über das Internet, ist eine Weiterleitung der Verbindung innerhalb des lokalen Netzwerks notwendig. Vergibt der jeweilige Internet Service Provider eine dynamische öffentliche IP Adresse, ist der Server des XPort nicht immer unter der gleichen Adresse verfügbar. Dies wird durch ein DynDNS oder DDNS System überdeckt, das Domain Name Einträge aktualisiert und die entsprechenden IP Adressen bereithält. Dazu muss aber ein DynDNS Client, ein Router, der über eine solche Funktion verfügt, oder eine Software, in regelmäßigen Abständen oder bei IP Wechsel die momentane dynamische IP an den DynDNS Server melden, um den Namensserver zu aktualisieren. Bis die neue IP allen Namensservern bekannt ist kann es einige Minuten dauern. Der Dynamische DNS Server vollzieht die Umwandlung der Host- oder Seitennamen in die maschinenlesbare Adresse. Damit ist der Server unter dem bei DynDNS angelegten Hostnamen verfügbar, unabhängig von der momentanen dynamischen IP Adresse.

Dem Router obliegt die Aufgabe, Verbindungen, die über externe Hosts an seiner öffentlichen Adresse eingehen, mittels des NAT Mechanismus (Network Address Translation) auf lokale IP Adressen zu übersetzen und umgekehrt. Die Quell- oder Ziel-IP der Datenpakete wird dabei je nach Verbindungsrichtung ersetzt. Durch die Definition von Ports für ein Port-Forwarding, das dem Router mitteilt, an welche Adresse die Anfrage an einem bestimmten Port weiterzuleiten sind, werden die Weichen für die mögliche Socketverbindung zwischen Anwendersoftware und Server gestellt. Um eine Verbindung zu ermöglichen, sollten hier auch die Firewallinstellungen des Routers angepasst und die jeweiligen Ports freigeschalten werden. Diese Ports bzw. deren Nummern stellen für bestimmte Protokolle, deren unterschiedliche Datenströme voneinander getrennt sind, Kommunikationsendpunkte dar.

Das Hypertext Transfer Protokoll verwendet Port 80. Um die Webseite des XPort aufzurufen, wird durch den Browser über TCP an Port 80 eine HTTP - GET Anforderung gesendet. Dieser Port muss durch den Router also weitergeleitet werden. Dass gleiche gilt für den Port 10001, an dem das Socket vom Java Applet zum Datenaustausch mit dem

XPort geöffnet wird. Am Port 30718 reagiert das XPort auf UDP Pakete. Ist bei Anbindung an das Netzwerk der DHCP Client des XPort aktiviert, sendet dieser Broadcasts mit Zieladresse 255.255.255.255 und Quelladresse 0.0.0.0 in Form von UDP Paketen, die die MAC Adresse beinhalten. Wird ein DHCP Server z. B. ein Router angesprochen, antwortet dieser mit einem UDP Paket, das die nötigen Netzwerkeinstellungen wie IP Adresse, Subnetzmaske, IP des DHCP Servers beinhaltet. Nach gegenseitigem Quittieren werden diese übernommen. Zusätzlich muss auch der Port 9999 weitergeleitet werden, über den das XPort eine Telnet Verbindung unterhalten kann. Damit sind Änderungen an dessen Konfiguration möglich.

Kapitel 5

Massenspeicher zur Langzeitaufzeichnung

Um Rückschlüsse zu ziehen, wie sich ein elektrischer Verbraucher am Netz verhält, ist es interessant, die gesammelten Messdaten über einen längeren Zeitraum in einem wählbaren Abtastintervall für eine spätere Auswertung am Computer zu speichern. Zur Verarbeitung empfiehlt sich ein Tabellenkalkulationsprogramm um die Datenfülle einerseits übersichtlich zu gestalten und andererseits um die Möglichkeit zu besitzen, diese weiter aufzubereiten sowie auch Diagramme zu gestalten. Um die Datenerfassung und die Verarbeitung sowohl zeitlich als auch örtlich möglichst unabhängig voneinander zu machen, bietet sich ein austauschbares Speichermedium, wie ein USB-Flash-Stick oder eine SD-Card, anstelle einer kabelgebundenen Direktübertragung von Mikrocontroller zu Computer an. Das Energiemessgerät soll also in diesem Punkt zu einem Datenlogger erweitert werden. Ein günstiges Flash-Speichermedium eignet sich in allen Belangen bestens dafür.

Die Übertragungsgeschwindigkeit hängt dabei sowohl vom Mikrocontroller, der für die Übertragung benutzten Schnittstelle, als auch von dem Flash Medium selbst ab, dessen Schreibgeschwindigkeit ohnehin langsamer als die Lesegeschwindigkeit ist und richtet sich nach dem langsamsten Glied dieser Kette. Sowohl SD als auch MMC Karten verfügen über einen SPI Modus, um Daten mit einem Mikrocontroller auszutauschen. Um diese aber auf einem Flash-Medium mit FAT-Filesystem zu speichern, damit sie dann auch am Computer erkannt und gelesen werden können, sind umfangreiche Softwarefunktionen nötig, damit der Mikrocontroller das FAT-Format handhaben kann. Da seine primäre Aufgabe aber in der Messdatenverarbeitung und Weiterleitung liegen sollte, ist eine ausgereifere und mehr angepasste Lösung zielführender. Darüber hinaus ist eine Flash Karte zwar klein und der SD Card Socket Connector im Gerät so zu verbauen, dass die Speicherkarte darin verschwindet, doch gehört der dazu benötigte Kartenleser am Computer nicht unbedingt zu einer Primärschnittstelle wie beispielsweise USB.

USB-Interfaces sind heute sowohl an billigen Flash Memory Speichern, als auch an vielen Peripherie-Geräten weit verbreitet, jedoch vornehmlich konzipiert für den PC Markt. Um die Möglichkeit einer USB-Schnittstelle in kleinen 8 oder 16 Bit Embedded Systems zu nützen, bedarf es eines USB-Host-Controllers, da es Mikrocontrollern einfach an Interfaces, Ressourcen und Performance mangelt, um mit einer USB-Schnittstelle direkt umzugehen.

Der VNC1L Vinculum Single-Chip Embedded-Host-Controller stellt ein passendes In-

terface zwischen einem Mikrocontroller und einem USB Port bereit und organisiert zusätzlich das FAT Dateisystem des verbundenen Flash Speichermediums. Damit lassen sich dann Dateien erstellen, in denen die Messwerte für den Benutzer lesbar und übersichtlich abgelegt werden können, wie z.B. *.txt oder *.csv Files, von denen aus die Daten dann einfach in ein Tabellenkalkulationsprogramm importiert werden können.

5.1 CSV Dateiformat

Im Dateiformat CSV (Comma Saperated Values), für das es keine einheitliche Spezifikation gibt, lassen sich einfach strukturierte ASCII Text-Daten in Form von Tabellen, in Zeilen und Spalten organisieren und so einfach zwischen verschiedensten Computersystemen und Applikationen austauschen. Die einfache Strukturierung der Daten innerhalb der Datei macht das Datenformat für Mikrocontrollersysteme, die ja über wenige Ressourcen verfügen besonders interessant. Dazu kommt noch der Vorteil, dass dieses Dateiformat durch Software wie beispielsweise Texteditoren unkompliziert bearbeitet, oder in Tabellenkalkulationen wie Excel, das das nicht einheitliche Format flexibel handhaben kann, importiert und aufbereitet werden kann.

In CSV-Dateien werden die einzelnen Datenfelder, also Spalten, mit einem Zeichen getrennt, meist Komma oder Semikolon und die Datensätze, also Zeilen, mit einem Zeilenumbruch. Dieser besteht aus zwei, einem Carriage Return (0x0D) und einem Line Feed (0x0A) Zeichen und wird ebenso wie das Semikolon vom Mikrocontroller zwischen den Daten eingefügt. Ein Feldbegrenzerzeichen wie das Doppelhochkomma wird nötig um diese Sonderzeichen selbst innerhalb der Daten nutzen zu können, welches wiederum seinerseits im Datensatz auftauchen kann, wenn es doppelt benutzt wird. Vor den eigentlichen Datensätzen kann ein Kopfdatensatz enthalten sein, der die Spalten mit einem Namen versieht. Weil kein einheitlicher Standard existiert, können die Systeme, die die Daten miteinander austauschen flexibel aufeinander abgestimmt werden.

Nach diesen simplen Grundregeln lassen sich somit die Datensätze, die Messwerte sowie dazugehörige Zeitwerte umfassen und vom Mikrocontroller bereitgestellt werden, einfach in einem tabellenorientierten Datenformat wie CSV abspeichern und verwalten. Da CSV bis auf Zeilentrennung und Spaltentrennung sonst keine speziellen zusätzlichen Zeichen und Formatierungen benötigt, werden auch die Dateigröße und der Bearbeitungsaufwand durch den Mikrocontroller minimiert.

Nach der Definition des spezifischen Trennzeichens wird beim Importieren der durch den Mikrocontroller erstellten CSV-Datei in Excel das Kommazeichen, welches beim Konvertieren von einer float Zahl zu einem ASCII String aus einem Punkt besteht, entsprechend gewählt. Im Anschluss werden die Zellen ihrem Typ (Text,Zahl,Zeit) entsprechend formatiert, sodass mit den Daten in Folge mathematische Operationen ausgeführt oder Diagramme erstellt werden können.

5.2 VNC1L

Der VNC1L Host Controller basiert auf einen Prozessor Kern mit zwei DMA-Controllern, die es erlauben Daten über das interne SRAM zwischen der USB Serial Interface Engine (SIE) und dem Mikrocontroller-seitigem Interface (SPI, UART, FIFO Parallel) auszutau-

schen. Damit wird der Prozessor entlastet und der Datentransfer beschleunigt. Die SIE bewerkstelligt die Konversion zwischen Daten des physikalischen USB Layers und des internen parallelen Bus. Ein numerischer 32-Bit-Co-Prozessor führt die arithmetischen Berechnungen für das FAT-File System durch. Dieses ist für den Benutzer völlig transparent. Die Kommunikation zur Steuerung durch den Mikrocontroller erfolgt über einfache DOS ähnliche Kommandos, die auch in einer verkürzten für Maschinen optimierten Fassung existieren.[FTD06b]

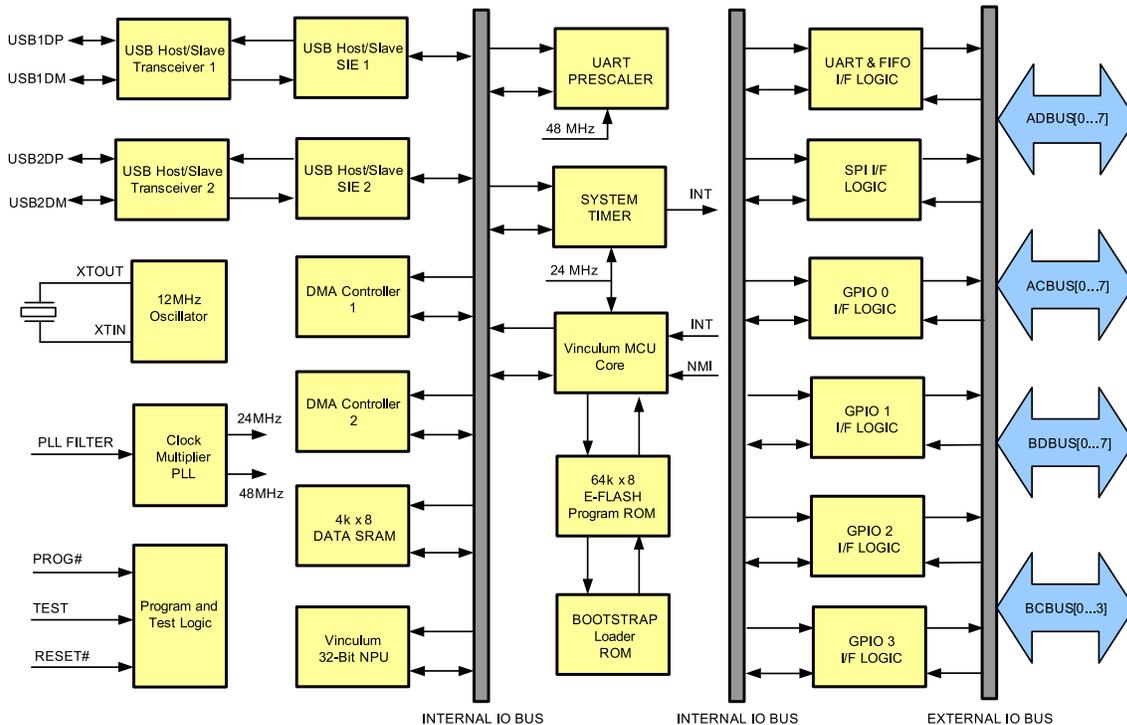


Abbildung 5.1: Blockschaltbild VNC1L[FTD06b].

Die Abbildung 5.1 zeigt den prinzipiellen Aufbau des VNC1L. Die USB-Transceiver sind das physikalische Interface und unterstützen USB 1.1 und USB 2.0 Standards sowie Low-speed als auch Full-speed Datenraten. Der VNC1L verfügt über zwei unabhängige USB Host/Slave Ports mit integrierten Pull-up und Pull-down Widerständen, an die abhängig von der installierten Firmware verschiedene USB Geräte angeschlossen werden können. Im konkreten Fall eines USB-Flash-Speichermediums, auf das Schreib- und Lesezugriffe ausgeführt werden sollen, muss der USB-Port 2 und die standardmäßig installierte Firmware VDAP verwendet werden.

An jedem Port stellt der Ausgangstreiber 3.3V Pegel für das Differenzsignal der beiden Datenleitungen bereit, während ein differenzieller Receiver und zwei single-ended Receiver die Datensignale empfangen. Auch das SE0 Signal (single ended zero; D+ und D- sind low), welches als Reset des Devices aufgefasst wird, wird unterstützt.

Der 12Mhz Takt des Oszillators wird über eine PLL verdoppelt und vervierfacht und stellt danach die Taktfrequenz der SIE Blöcke, des MCU Kerns und der Timer dar. Dieser 8-Bit-MCU hat eine Harvard Prozessor Architektur mit separaten Code und Daten Speicher und verwendet einen enhanced CISC Befehlssatz, was schnelle Ausführung und

Codekomprimierung ermöglicht. Die Betriebsspannung beträgt 3.3V. Um Geräte wie USB Flash Speicher, die über den Bus versorgt werden, zu betreiben, sind zusätzlich noch 5V an Spannung nötig, die direkt an die Schnittstelle geführt werden. Die I/O Pins sind 5V tolerant was eine direkte Verbindung mit dem Mikrocontroller ohne die Verwendung eines Level Shifters ermöglicht. Die maximale Leistungsaufnahme bei einer maximalen Betriebsspannung von 3.6V ist mit 250mW angegeben, wovon maximal 200mA dem Gerät am USB-Anschluss zur Verfügung gestellt werden und etwa 25mA dem VNC1L selbst. In der Praxis hängt die tatsächliche Leistungsaufnahme also vom angeschlossenen Gerät ab.

Der im 48 Pin LQFP Package ausgelieferte IC bedarf keiner umfangreichen externen Beschaltung. Mit dem 12MHz Quarz mit zwei Lastkondensatoren auf Masse, einigen Pull-up und Pull-down Widerständen an bestimmten Leitungen zur Festsetzung gewisser Zustände und Funktionen, sowie einigen Entstör- und Bufferkondensatoren und 27 Ω Serienwiderständen in den USB Datenleitungen ist die Beschaltung komplett.

5.2.1 VDIP1 Modul

Der Hersteller liefert zur Integration in bestehende Projekte auch kleine Leiterplatten mit dem VNC1L Host Controller und der, für den Betrieb notwendigen Schaltung an Bauelementen, samt USB A Type Stecker. Das Board bietet über zwei Stiftleisten Zugang zu allen wichtigen Interface Pins und passt in einen 24 Pin DIP Sockel. Zur freien Wahl der Schnittstelle ermöglichen zwei Jumper die Konfiguration des gewünschten I/O Interfaces, mittels der entsprechenden Kombination von Pull-up oder Pull-down Widerständen an den Leitungen ACBUS5 und ACBUS6. Folgende Kombinationen von Jumpern und damit selektierbaren I/O Schnittstellen sind möglich.

ACBUS 6 (VNC1L Pin 47)		ACBUS 5 (VNC1L Pin 46)		I/O Mode
Pull Up	J4 1-2	Pull Up	J3 1-2	Serial UART
Pull Up	J4 1-2	Pull Down	J3 2-3	SPI
Pull Down	J4 2-3	Pull Up	J3 1-2	Parallel FIFO
Pull Down	J4 2-3	Pull Down	J3 2-3	Serial UART

Tabelle 5.1: Konfiguration Schnittstelle VDIP1 bzw. VNC1L [FTD06a].

Die Betriebsspannung des Moduls beträgt 5V, wobei diese nur für den Betrieb des Flash Speichermoduls am USB-Port nötig ist und der USB-Host-Controller selbst an 3.3V mittels eines vorhandenen Spannungsreglers betrieben wird. Der Schaltplan des VDIP1 Moduls findet sich im Anhang.

5.2.2 Schnittstellen

Die Möglichkeit der wählbaren Schnittstelle, SPI, UART oder FIFO Parallel stellt einen großen Vorteil dar, da der USB-Host-Controller damit an die bereits gegebene Prozessumgebung angepasst werden kann. Der Mikrocontroller XC167 seinerseits bietet eine Vielzahl an implementierten Interfaces, sodass als Schnittstelle für diese Aufgabe der noch unbelegte zweite SPI Bus gewählt wurde. Dieser ist im Vergleich zu der UART wesentlich schneller und benötigt im Vergleich zur parallelen Schnittstelle weniger Leitungen. Da der

XC167 in der verwendeten Applikation keinen Mangel an unbelegten Input bzw. Output Pins hat, kann durchaus über die Realisierung mit einem Parallelen Interface nachgedacht werden, an dem ein ganzes Byte näherungsweise in der gleichen Zeit wie ein Bit am SPI Bus übertragen werden kann. Zu bedenken ist, dass dann dieses Byte erst im Buffer des VNC1L steht und von dort aus seriell über die USB-Schnittstelle auf das Flash Speichermedium übertragen werden muss. Dies kann zur Folge haben, dass es keinen effektiven Geschwindigkeitsvorteil gegenüber SPI gibt, da Daten zwar schnell in den Buffer geschrieben werden, woraufhin dieser sich aber schneller füllt, als die Daten weiter übermittelt werden können und es so zu einem Stau kommt, der dann zwischenzeitige Wartephase zur Auflösung erforderlich macht. Um eine freie Wahl des Interfaces auch in der fertigen Applikation zu ermöglichen wurde das VDIP-Board sowohl über SPI Schnittstelle, als auch über, die für die parallele Übertragung notwendigen Leitungen mit dem Mikrocontroller verbunden. Zu beachten ist, dass die physikalische Verbindung der SPI Leitungen erst über drei Lötjumper zustande kommt, wogegen die parallelen Leitungen fix verbunden sind. Bei der Programmierung ist somit Augenmerk auf die nicht gebrauchten Leitungen beziehungsweise auf die, sowohl vom SPI als auch vom parallelen Interface verwendeten Leitungen zu legen. Die Pins des Mikrocontrollers an denen diese Leitungen anliegen müssen auf Input geschaltet sein, um mögliche Kurzschlüsse aufgrund von Software zu vermeiden.

5.2.3 SPI des VNC1L

Der SPI Bus umfasst die zwei Datenleitungen, die Taktleitung und eine CS-Leitung. Die maximale Taktfrequenz an der CLK Leitung kann bis zu 12MHz betragen. Im Betrieb mit dem XC167 wird in der Praxis eine Baudrate von 10 MBaud gewählt, was einer Bitdauer von 100ns an den Datenleitungen entspricht. Das MSB wird als Erstes übertragen, der idle Zustand des Taktsignals ist low, Daten werden mit der fallenden Flanke des CLK Signals ausgetaktet und mit der steigenden Flanke übernommen.

Die übermittelte Datengröße pro Übertragungszyklus beträgt 12 Bit und beinhaltet eine generelle 1 zu Beginn, die als Startsignal anzusehen ist, gefolgt von einem Bit, das aussagt, ob es sich um einen Lese- oder Schreibzyklus handelt, einem Adressbit, welches 0 für das Datenregister als Ziel ist, 8bit Daten und einem abschließenden Statusbit, das der VNC1L sowohl im Lese als auch im Schreibzyklus an den Mikrocontroller übermittelt.

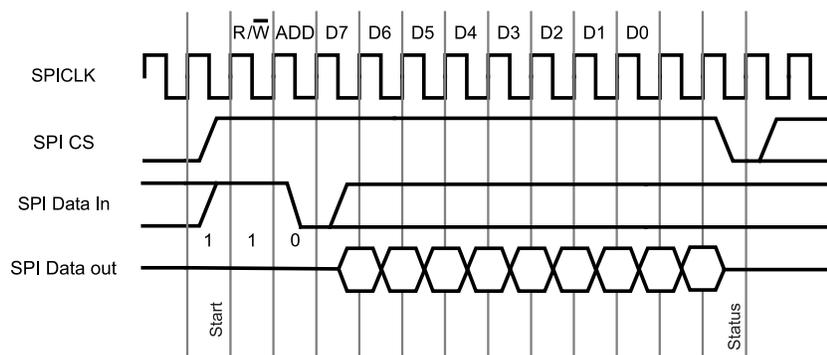


Abbildung 5.2: Lesezyklus SPI VNC1L[FTD06b].

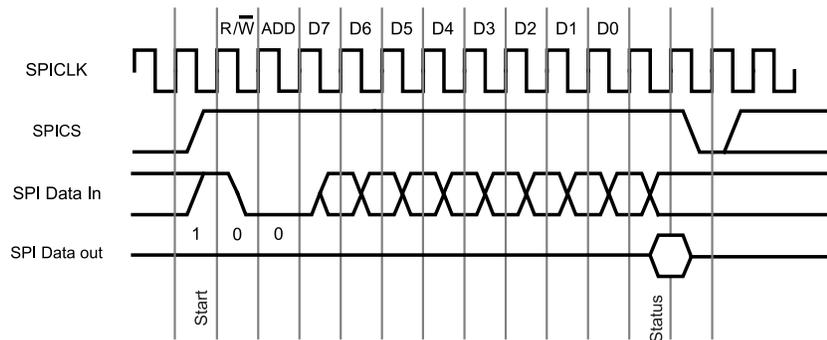


Abbildung 5.3: Schreibzyklus SPI VNC1L[FTD06b].

Die Abbildungen 5.2 und 5.3 zeigen einen Lese- und einen Schreibzyklus und die Signalzustände an den Leitungen. Zu Beginn wird durch das High setzen der CS Leitung der Baustein ausgewählt. CS muss während des ganzen Schreib- oder Lesezyklus High sein, damit Daten korrekt gesendet oder empfangen werden, sowie für zumindest eine Taktperiode des Taktsignals nach dem Transfer, bevor der Nächste startet. An den Daten-In und -Out Leitungen folgen die Bits im bereits erwähnten Muster. Das Ende bildet das Statusbit für den eben durchgeführten Transfer und sollte überprüft werden, da es Auskunft über diesen gibt. Im Lesezyklus bedeutet eine Null, dass die vom VNC1L übermittelten Daten Neue im Vergleich zum vorigen Lesezyklus sind. Eine Eins signalisiert, dass es sich um alte Daten handelt und der Lesevorgang wiederholt werden muss. Eine Null im Schreibzyklus zeigt an, dass das Byte vom VNC1L übernommen wurde, wohingegen eine Eins einem vollen Buffer entspricht und der Vorgang wiederholt werden muss. Der Mikrocontroller überwacht dieses Statusbit und führt gegebenenfalls die vorangegangene Aktion erneut aus. Neben dem Daten Register des VNC1L kann über SPI Schnittstelle auch das Statusregister angesprochen werden (ADD=1). Der Hersteller gibt hierfür aber wenig konkrete Informationen in Bezug auf die SPI Schnittstelle an. Aufgrund von Beobachtungen lässt sich lediglich auf die vermeintliche Funktionsweise schließen. Vermutlich soll mit der Abfrage der Bits des Statusregisters eine Art softwaremäßiges Handshaking implementiert werden können, wie es beim UART und Parallelen Interface möglich ist. Die ersten zwei Bits dieses Registers sind für den Status der Buffer relevant und werden mit RXF# (Receive Buffer Full) und TXE# (Transmit Buffer Empty) bezeichnet. Im idle Zustand des Chips ergibt sich immer der Registerwert von 0x12. Das bedeutet, dass das TXE# Bit gesetzt ist, da keine neuen Bytes zum Abholen vorliegen. Ist RXF# high ist das ein Verweis darauf, dass momentan keine weiteren Daten durch den VNC1L aufgenommen werden können. Da es durchaus möglich ist, dass der FIFO-Buffer voll ist, während Daten noch an Selbigen übertragen werden, kann es bei den über die USB-Schnittstelle übermittelten Daten zu einem Informationsverlust kommen. Dieser Fall ist am wahrscheinlichsten bei der Durchführung von Schreibzugriffen auf einem Flash Speicher beim Erreichen der Sektorgrenze, wenn die FAT durch den VNC1L bearbeitet werden muss und keine Zeit für die Weiterleitung der Daten über, die USB Schnittstelle besteht. In der Praxis hat sich gezeigt, dass beim schnellen kontinuierlichen Schreiben anders als beim Schreiben in 512Byte Blöcken, Daten jeweils an den Sektorgrenzen verschluckt werden. Somit wäre es vorteilhaft die Sektorgrenze feststellen zu können bzw. an selbiger zu bestimmen ob der

VNC1L zur Aufnahme weiterer Daten bereit ist.

Wenn vor jedem Datentransfer erst das RXF# Bit des Statusregisters geprüft wird und dieser nur ausgeführt wird, wenn RXF# Low ist, kommt es zu keinem Datenverlust, da somit überprüft wird, ob die FIFO für weitere Datenaufnahme bereit ist. Der Mikrocontroller muss dann gegebenenfalls den Transfer unterbrechen. Anzumerken ist, dass man so aber die effektive Schreibgeschwindigkeit durch das kontinuierliche zwischenzeitliche Lesen halbiert.

5.2.4 Firmware

Für die verschiedensten Arten von Geräten, die am USB-Port des VNC1L angeschlossen werden können, existieren unterschiedliche Versionen von Firmware, die ungleiche Funktionen wahrnehmen. Die standardmäßig am VNC1L installierte Firmware, VDAP (Vinculum Disk and Peripheral) interpretiert eine fixe, zur Verfügung stehende Auswahl an Kommandos, die über das Interface übertragen werden, und führt die entsprechenden Aufgaben aus. Diese Version der Firmware erlaubt den Anschluss eines USB-Flash-Speichers an Port 2 und eines anderwertigen USB Slave Gerätes an Port 1 (vgl. Abb. 5.4).

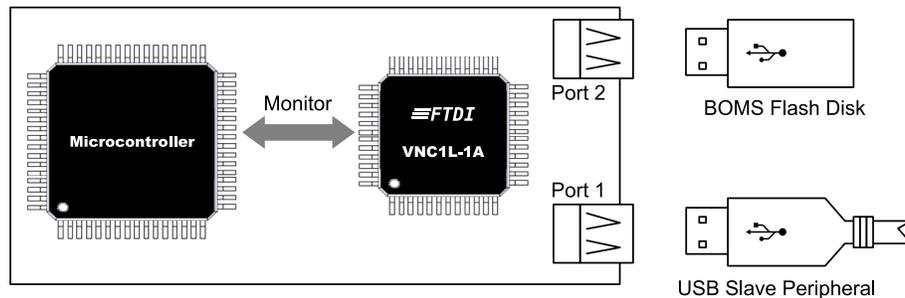


Abbildung 5.4: Command Monitor VNC1L [FTD08].

Die Kommunikation und die Steuerung des VNC1L erfolgen über DOS ähnlichen Kommandos, die an den Command Monitor gesendet und von diesem ausgeführt werden. Der Monitor stellt somit ein Interface auf Softwareebene dar. Dabei wird zwischen zwei Betriebsarten unterschieden, Command Mode, in dem Kommandos durch den Monitor interpretiert und ausgeführt werden und Data Mode, in dem alle Informationen, die an der Schnittstelle eingehen, direkt an die USB-Schnittstelle weitergeleitet werden. Es gibt zwei Eingabemöglichkeiten der Kommandos, Extended und Short. Im Extended Modus werden druckbare Zeichen verwendet und die Kommandos bestehen aus mehr Zeichen, wogegen im Short Modus binäre Werte verwendet werden, weshalb dieser somit besser zur Interaktion mit einem Mikrocontroller geeignet ist. Standardmäßig befindet sich der Monitor nach einem Reset im Extended Modus.

Der Command Monitor des VNC1L reagiert auf ein Kommando, indem er nach der Ausführung ein Prompt zurücksendet und damit anzeigt, dass er bereit für das Nächste ist, oder eine Error Nachricht und damit signalisiert, dass das Kommando nicht ausgeführt werden konnte oder nicht existiert. Diese Prompts oder Error Nachrichten werden stets durch ein Return Zeichen (0x0D) komplettiert. Ein erfolgreich ausgeführtes Kommando wird also im SCS (Short Command Set) Modus mit „>0x0D“ quittiert. Des Weiteren generiert der VNC1L auch Events, die beispielsweise Auskunft darüber geben, ob ein

Gerät angesteckt oder entfernt wurde. Die synchrone SPI-Schnittstelle bringt dabei im Gegensatz zu der asynchronen UART einige Besonderheiten im Softwaredesign mit sich, aufgrund der Tatsache, dass sowohl immer gleichzeitig Daten gesendet als auch empfangen werden. Wenn Events im VNC1L auftreten, wie beispielsweise beim Anstecken eines USB-Gerätes, wird eine Meldung erzeugt, die nicht unmittelbar wie an der asynchronen Schnittstelle übermittelt und im Mikrocontroller mittels Receive Interrupt erkannt werden kann, sondern es bedarf der regelmäßigen Abfrage des SPI Bus durch den Mikrocontroller ob eines aufgetretenen Events. Der VNC1L verfügt über Kommandos, mit denen Daten auf den USB-Flash Speicher gelesen und geschrieben werden können. Diese Disc Commands managen das Dateisystem und ermöglichen die Interaktion mit einem Flash Speicher, das Erstellen eines Files, Schreiben in Selbiges, Schließen des Files sowie Löschen, aber auch die Erstellung von Verzeichnissen.

5.2.5 Datenspeicherung

Für die Speicherung von Messdaten muss zuerst eine Datei, in welche gespeichert wird, mit dem Kommando OPW (Open for writing) gefolgt von dem Dateinamen geöffnet oder erstellt werden. Dafür sendet man über die gewählte Schnittstelle zum VNC1L entweder diese Sequenz („OPW Dateiname“) gefolgt von einem Return Zeichen oder wie im Short Command Set (0x09 0x20 Dateiname 0x0D), das für Mikrocontroller durch geringeren Zeichenbedarf gebräuchlicher ist. Wird die Datei des gewählten Formates erfolgreich erstellt, bestätigt dies der VNC1L durch die Aussendung einer Erfolgsmeldung. Der Befehl OPW erstellt nicht nur eine neue Datei vom spezifizierten Namen und Type, sondern öffnet diese Datei, falls sie bereits im Verzeichnis existiert. Die Dateizeit könnte ebenfalls im Zuge dieses Befehls auf das gewünschte Datum und Uhrzeit eingestellt werden. Nun da die Datei erstellt bzw. offen für die Bearbeitung ist, kann in diese mittels des WRF (write to file) Kommandos geschrieben, und auch von dieser gelesen werden (RDF). Es kann nur eine Datei zur gleichen Zeit geöffnet sein und bearbeitet werden. Ein Datei Pointer verwies auf die Stelle, an der ein nachfolgender Schreibbefehl Daten ablegt bzw. auf die Daten, die als Nächstes von einem Lesebefehl übernommen werden. Mit dem SEK Kommando kann der Datei Pointer innerhalb der Datei an den angegebenen Bytewert gesetzt werden. Es gibt nur die Möglichkeit dieser absoluten Positionsbestimmung des Datei Pointers. Wird eine Datei geöffnet, die schon Informationen enthält, werden diese nicht überschrieben sondern es wird am Ende der Datei mit dem Schreibvorgang begonnen, da der Datei Pointer auf das Ende zeigt, wenn diese Datei durch den VNC1L erzeugt wurde. Zum Abschluss muss die erstellte und bearbeitete Datei noch geschlossen werden (CLF). Erst durch diesen Vorgang wird die Datei, durch die Fertigstellung der FAT, physikalisch sichtbar und kann am Computer geöffnet und bearbeitet werden.

Wird die durch den VNC1L erstellte Datei nicht durch das CLF Kommando ordnungsgemäß geschlossen, wie zum Beispiel durch das vorzeitige Abziehen des Speichermediums, ohne den Schreibvorgang durch Tastendruck zu beenden, wird die Dateigröße in der Dateiverzeichnistabelle nicht aktualisiert und die Datei ist am Computer nicht als solche auszumachen bzw. sichtbar. Dies ändert aber nichts an der Tatsache, dass die soeben übermittelten Daten am USB-Flash Speicher abgelegt sind. In diesem Fall lässt sich mithilfe eines geeigneten Hex Editors das Speichermedium nach einem signifikanten Bit Muster innerhalb der gesicherten Messdaten wie zum Beispiel der Kopfzeile durchsuchen und mit

den vorhandenen Daten eine eigene Datei erstellen.

5.2.6 Beschränkungen

Folgende Beschränkungen müssen aufgrund der VNC1L Kompatibilität gemacht werden. Alle verwendeten Flash-Speicher-Medien müssen eine Sektorgröße von 512 Byte haben, was auf die meisten handelsüblichen Produkte zutrifft. Stellt der VNC1L fest, dass die Sektorgröße nicht 512 Byte beträgt, ist der Zugriff auf das FAT-Dateisystem nicht möglich. Weiters werden nur USB Speicher die den SCSI oder den AT Befehlssatz unterstützen angenommen.

Nur die Dateiformate FAT12, FAT16, welches besser bekannt ist unter dem Namen FAT und FAT32 werden bis dato von der Firmware des Vinculum unterstützt. Vor dem Anstecken eines USB-Flash-Speichers ist deshalb darauf zu achten, dass dieser dementsprechend formatiert ist. Die FAT (File allocation Table) ist eine Tabelle fester Größe, die die freien bzw. belegten Cluster des Dateisystems verzeichnet. Ein Cluster besteht aus einem oder mehreren Sektoren und ist die Zuordnungseinheit, die von einer Datei belegt wird. Die FAT beinhaltet die Informationen zu allen Clustern des Speichermediums. Die Wahl des FAT-Dateisystems hängt mit der Größe des Speichermediums zusammen. FAT16, das allgemein nur mehr als FAT bezeichnet wird, hat einen Adressraum von 16Bit was 65536 Einträge möglich macht, allerdings nur 512 im Rootverzeichnis. Mit einem 8Bit Zähler, der die Anzahl der Sektoren pro Cluster bestimmt, ist die Anzahl der Sektoren pro Cluster auf 64 limitiert. Mit der Standard Sektorgröße von 512Bytes und 64 Sektoren pro Cluster ergibt sich eine Clustergröße von 32KB und weiters eine mögliche Partitionsgröße von 2GB. Dies beschränkt die Größe des verwendbaren USB-Flash-Speichermediums.

In der Praxis hat sich außerdem gezeigt dass trotz eines 32 Bit Co-Prozessors, FAT16 durch den kleineren Adressraum schnellere Zugriffszeiten auf den Speicher ermöglicht, was bei der kontinuierlichen Übertragung von Daten zu weniger Problemen bezüglich eines Datenstaus und einer besseren Performance führt. Aus diesem Grund ist es dringend zu empfehlen das FAT16 Dateiformat vorzuziehen, um keine Datenkollisionen zu verursachen.

Außerdem dauert der Prozess der Auffindung von freien Sektoren und Clustern länger, je weniger verfügbaren Speicherplatz das eingelegte Volumen aufweist, womit die Zugriffszeiten gesteigert werden. Beim FAT Dateisystem werden Cluster beschrieben, wie sie gerade gebraucht werden, was zur Fragmentierung führt und weiter zu Verzögerungen beim Schreiben, durch diese nicht sequenzielle Anordnung. Diese Verzögerungen nehmen mit der Auslastung des Speichermediums zu. Deshalb ist bei einem solchen Medium eine vorherige Defragmentierung empfehlenswert.

Kapitel 6

Gesamtschaltung

Nach umfassender Beschreibung der wesentlichen Komponenten, die zur Erfüllung der Gesamtfunktion zusammenarbeiten, deren Eigenschaften und Funktionsweise sowie der Schnittstellen zur Verbindung untereinander, folgt deren Zusammenführung zu einer Gesamtschaltung bzw. einem System. Unter Einbeziehung einiger weiterer Elemente und gewisser Aspekte im Zusammenhang mit dem Schaltungsentwurf, auf die nachfolgend eingegangen werden sollen, ergibt sich die Schaltung, deren Schaltplan und Layout im Anhang zu finden sind.

6.1 Benutzerschnittstellen Ein- und Ausgabe

Die direkte Möglichkeit zu einer Eingabe und damit zur Steuerung des Systems besteht für den Benutzer mittels des vorhandenen Tastenfeldes. Als unmittelbare Ausgabeinheit verfügt das Gerät über ein LC Display.

6.1.1 Schaltung zur Entprellung der Tasten

Um nicht durch kontinuierliche Statusabfragen und in Software implementierten Entprellmechanismen Rechenzeit zu verlieren, sowie dem Konzept einer Ereignis-gesteuerten Softwarearchitektur zu folgen, wird durch die fallende Flanke eines Tastendrucks ein Interrupt erzeugt und der Tastendruck als solcher registriert. Da wegen des Prellvorgangs des Tasters dieser pro Tastendruck mehr als einmal detektiert werden kann, ist eine Schaltung zur Entprellung notwendig. Da je nach Art des verwendeten Schalters der Prellvorgang unterschiedlich lang ausfallen kann, wurde auf der Leiterplatte die Möglichkeit zur Realisierung verschiedener Entprellschaltungen geschaffen, die auf dem Prinzip der Integration mittels eines Tiefpasses beruhen.

Durch die Wahl einer entsprechenden Zeitkonstante gegeben durch den Kondensator C und den Entladewiderstand R_2 sollte beim Entladen des Kondensators stets ein eindeutiger Pegel frei von Prelleffekten am digitalen Eingang des Mikrocontrollers anliegen. Die Flankensteilheit kann dabei gegebenenfalls mit einem Schmitt-Trigger erhöht werden. Der Einsatz einer Diode wie in Schaltung (Abb. 6.1c) ermöglicht eine größere Zeitkonstante und eine längere Einstellzeit bei geschlossenen Schaltern sowie eine kleinere Aufladezeit des Kondensators. Im Gegensatz dazu Schaltung (Abb. 6.1b), bei der die Aufladezeit durch

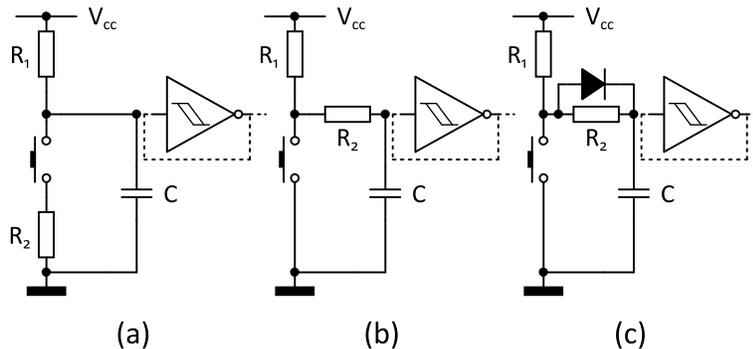


Abbildung 6.1: Schaltungen zur Entprellung der Tasten

den Widerstand R_2 steigt, was somit die Schaltung verlangsamt, da die Spannung langsam einen Bereich passiert der keinen eindeutigen Logikpegel zuzuordnen ist. In Schaltung (Abb. 6.1a) können aufgrund des, bei geschlossenem Schalter entstehenden Spannungsteilers, nur entsprechend kleine Werte für R_2 gewählt werden, soll das Spannungsniveau noch als Logikpegel low gelten.

6.1.2 LC Display

Die Ausgabe der Messergebnisse erfolgt über ein 4x20 LC Display mit einem HD44780 Displaycontroller oder einem kompatiblen wie dem KS0066. Um I/O Leitungen einzusparen, wird das Display im 4-Bit Modus betrieben, der nur die oberen vier Datenleitungen verwendet (DB4-DB7). Um ein Byte zu übertragen, braucht man somit zwei Zugriffe, wobei zuerst das höherwertige Nibble (= 4 Bit), Bit 4 bis Bit 7 übertragen wird und dann das niederwertige, Bit 0 bis Bit 3. Zur Kontrastregelung wird ein $10k\Omega$ Potenziometer zwischen Versorgung und Masse geschaltet und der mittlere Anschluss als Kontrastspannung (V_0) verwendet. Alternativ kann V_0 auch auf Massenpotenzial gelegt werden, woraus höchstmöglicher Kontrast resultiert. Zusätzlich verfügt das Display zur Beleuchtung über eine LED, deren Anodenanschluss über einen Vorwiderstand von 80Ω an die Versorgungsspannung und deren Kathode auf Masse gelegt wird. Zum Stromsparen kann die Beleuchtung auch durch den Mikrocontroller über einen Transistor, der als Low-Side Switch verwendet wird gesteuert werden. Nach dem Power-up muss das Display für den Betrieb erst vom Mikrocontroller nach einer fixen Sequenz initialisiert werden (siehe Display Datenblatt).

6.2 Spannungsversorgung und galvanische Trennung

Ein DC/DC-Wandler ist notwendig, der die Messseite von der Mikrocontrollerseite auch über die Versorgungsleitungen galvanisch trennt, da die beiden Schaltungsteile unterschiedliche Potenziale aufweisen, zumal die Schaltung nicht direkt aus dem zu messenden Netz versorgt wird, sondern über eine zum Betrieb erforderliche Gleichspannung durch eine separate DC Versorgungsquelle. Der vorliegende DC/DC-Wandler TEN4-2423 besitzt dabei eine Isolationsspannung von $1.5kV_{dc}$ und einen Isolationswiderstand von $> 1000M\Omega$. Außerdem verfügt er über einen großen möglichen Eingangsspannungsbereich von 9-36Volt

(4:1), damit die Schaltung sowohl an 12V als auch an 24V betrieben werden kann, sowie über eine positive und negative Ausgangsspannung von 15V bezüglich des Massenpotenzials, welche für die Versorgung der Hallsensoren nötig ist. Der Fehler in der Ausgangsspannung des DC/DC Konverters liegt bei $\pm 1\%$. Die Hallsensoren arbeiten um den Bereich von $\pm 5\%$ der Betriebsspannung von $\pm 15\text{V}$ gemäß ihrer Spezifikation. Der maximale Ausgangsstrom des DC/DC-Wandlers an den Pins beträgt $\pm 130\text{mA}$. Das entspricht somit einer Ausgangsleistung von 4Watt. Der Eingangsstrom liegt bei Vollast typischerweise zwischen 200 und 400mA, abhängig von der Eingangsspannung. Die maximale kapazitive Last an den beiden Ausgängen beträgt $300\mu\text{F}$, worauf bei der Wahl der Stützkondensatoren bzw. Glättungskondensatoren, zur Minimierung des leichten Ausgangs-Ripple von typischen 50mV geachtet werden muss, ebenso wie auf die Nennspannung dieser Kondensatoren.

Die $\pm 15\text{V}$ Ausgangsspannung des DC/DC-Wandlers werden den Hallsensoren als Betriebsspannung bereitgestellt (typische Stromaufnahme pro Sensor $< 15\text{mA}$). Die Betriebsspannung des restlichen Messteiles beträgt 5V und wird über einen, dem DC/DC-Wandler nachgeschalteten linearen Spannungsregler eingestellt.

Die Versorgung der weiteren Schaltungsgruppe des Mikrocontrollerteils übernimmt ein einstellbarer Schaltregler, um größere Verlustleistungen durch Wärme, aufgrund des großen Eingangsspannungsbereiches der Gesamtschaltung zu vermeiden. Wegen der schnellen Schaltvorgänge und auftretenden elektro-magnetischen Störungen muss, zur Minimierung dieser Effekte, dem Layout der Platine besondere Beachtung geschenkt werden. Es ergibt sich auch ein höherer Bauteilaufwand als bei linearen Längsreglern. Betrachtet man die erforderlichen Betriebsspannungen der Schaltungskomponenten, sind mit 2.5V für den Prozessorkern und die interne Logik des XC167, 3.3V für das XPort und 5V für die weiteren Komponenten drei Spannungsniveaus erforderlich.

Der Hersteller der Mikrocontroller empfiehlt die Nutzung eines geeigneten linearen Spannungsreglers TLE7469, der speziell dafür ausgelegt ist und sowohl die 5V für die Portstruktur als auch die 2.5V für die prozessorinterne Architektur einstellt, sowie die Reset Steuerung übernimmt. So soll sichergestellt werden, dass der Reset Pin low bleibt, bis die Einschalttrampe der Versorgungsspannung 2,25V an Vddi und 4,5V an Vddp erreicht hat, damit das System richtig gestartet wird. Der TLE7469 verfügt zusätzlich über einen internen Pull-Up Widerstand am Reset Ausgang, sodass der Baustein direkt mit dem Mikrocontroller verbunden werden kann. Der Spannungsregler hat zwar einen maximalen Eingangsspannungsbereich von bis zu 45V, doch ist schon bei den tatsächlich verwendeten Eingangsspannungen von 12V und 24V eine erhebliche Wärmeentwicklung aufgrund der Verlustleistung gegeben, sodass entsprechende Kühlung erforderlich ist. Aufgrund der Tatsache, dass der Mikrocontroller zusammen mit diesem Regler und der nötigsten Beschaltung auf einer Kernplatine mit geringen Dimensionen sitzt, die in die Schaltung integriert wird, ist hier eine ausreichende Kühlung nicht denkbar. Deswegen sind nur niedrige Spannungsniveaus am Eingang des Spannungsreglers möglich. Da es sich um keinen Low-Drop Spannungsregler handelt, ist eine Differenz zwischen der Eingangsspannung und der gewünschten Ausgangsspannung nötig (1,5-3V). Diese Spannung wird über einen verlustarmen Schaltregler am Eingang der Gesamtschaltung eingestellt.

Mit dem monolithischen 2A Step-Down Schaltregler LT3500 lässt sich zusätzlich zu der, über ein Widerstandsverhältnis bestimmbar, flexiblen primären Ausgangsspannung über einen integrierten Linearregler ein zweites Spannungsniveau einstellen. Das primäre

Spannungsniveau wird so gewählt, dass die Differenz zwischen Eingang und Ausgang der nachfolgenden Linearregler für deren Betrieb groß genug wird, was etwa einer Spannung von 7V entspricht. Durch den internen Linearregler des LT3500 wird außerdem ein zweites Spannungsniveau von exakt 3.3V eingestellt. Die Abbildung 6.2 veranschaulicht diese Versorgungsstruktur.

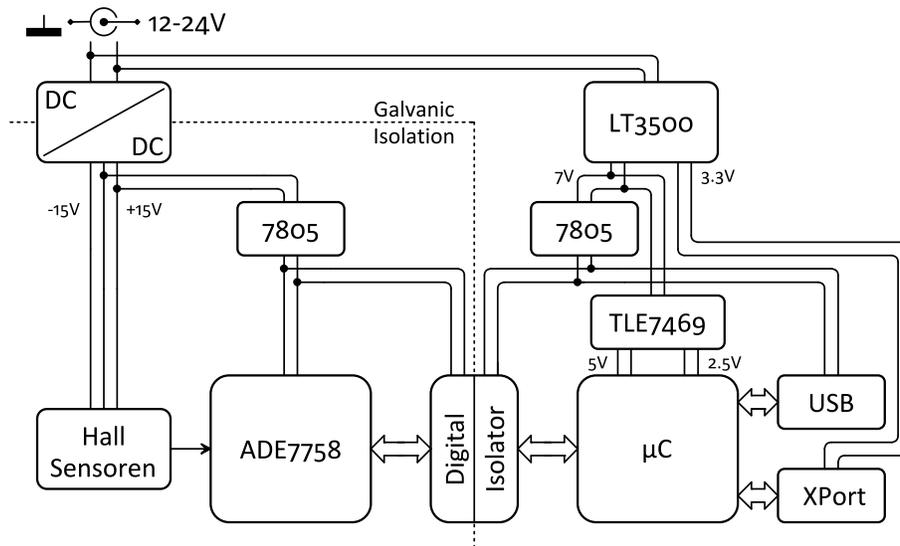


Abbildung 6.2: Versorgungsstruktur

6.2.1 LT3500

Beim LT3500 handelt es sich um einen monolithischen einstellbaren Step-down Schaltregler mit einem großen Eingangsspannungsbereich von maximal 40V und einem zusätzlichen linearen Längsregler, für die Einstellung eines zweiten Spannungsniveaus. Der dabei erreichbare Ausgangsstrom von maximale 2A, als Summe der beiden Ausgänge, hängt von der Eingangsspannung, der Wahl der richtigen Induktivität und der Schaltfrequenz ab. Die Ausgangsspannungen werden über Widerstandsteilerverhältnisse eingestellt (vgl. Abb. 6.3). Durch externe Beschaltung mit einem NPN Transistor fungiert der LT3500 als Regler für einen so konstruierten Low-drop Spannungsregulator. Die Schaltfrequenz wird ebenfalls durch die Beschaltung mit einem externen Widerstand zwischen den Grenzen 250KHz und 2.5MHz eingestellt. Je höher die Schaltfrequenz gewählt wird, umso kleiner kann die Drossel, die die Energie im magnetischen Feld speichert, werden, aber umso geringer darf die maximale Eingangsspannung für korrekten Betrieb aufgrund von Schaltverlusten sein. Bei der Wahl des Widerstandes von 20k Ohm folgt laut des Diagramms der Performance Charakteristik des Datenblattes ungefähr eine Schaltfrequenz von 1.75MHz. Die minimale Einschaltzeit des Schalters steigt, wie dort ebenso ersichtlich ist, mit der Temperatur an und damit gleichfalls der Duty-Cycle. Somit verringert sich bei steigender Temperatur die maximale Eingangsspannung. Bei einer eingestellten Ausgangsspannung von ca. 7V, um den folgenden linearen Spannungsreglern die nötige Spannungsdifferenz zu deren Ausgangsspannungen zu bieten, einem Spannungsabfall von 0.4V am internen Schalter und einer Vorwärtsspannung von 0.6V an der Freilaufdiode sowie einer Temperatur von

circa 80°C und der daraus resultierenden minimalen Zeit des geschlossenen Schalters, erreicht die maximale Eingangsspannung laut Formel weniger als 24 Volt. Dieser Umstand ist für den Betrieb zu beachten und bei höheren Temperaturen eventuell eine niedrigere Eingangsspannung zu wählen. Die Bestimmung der richtigen Induktivität erfolgt aus dem Festsetzen des Eingangsspannungsbereiches, der Ausgangsspannung und der Schaltfrequenz. Weiters wird diese Entscheidung auch vom Duty-Cycle und vom Strom-Ripple am Ausgang bestimmt. Ist die Drossel unterdimensioniert, nimmt man einen größeren Strom-Rippel in Kauf. Der Ausgangskondensator muss für eine entsprechende Glättung der Ausgangsspannung sorgen und darf deswegen nicht zu gering dimensioniert ausfallen. Wegen der vielen Parameter, die es zu berücksichtigen gilt und die voneinander anhängig sind, ergeben sich für die vorliegende Dimensionierung der Schaltung nur für eine Eingangsspannung von 12V ideale Verhältnisse am Ausgang .

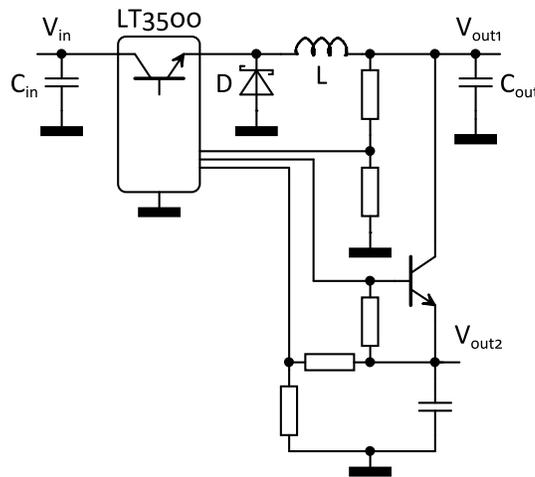


Abbildung 6.3: Prinzipschaltbild Schaltregler

Aufgrund der großen Schaltfrequenzen und der hohen Stromänderungen muss auf das Platinenlayout besonders Augenmerk gelegt werden, um Störungen zu vermeiden. Die großen Schaltströme fließen dabei durch den Schalter, die Freilaufdiode und den Eingangskondensator. Um induzierte Störungen zu vermeiden, sollte die Schleife die diese Komponenten aufspannen so klein wie möglich sein. Weiters sollte der gesamte Schaltungsblock durch eine umgebende Massefläche vom restlichen Schaltungsgebilde abgeschirmt werden.

6.2.2 Digital Isolator

Die galvanische Trennung der Messseite von der restlichen Schaltung ist nötig, um die Potenziale der beiden Schaltungsteile und die Spannungsversorgung vom Stromkreis der zu messenden Spannung zu trennen. Das gilt sowohl für die Versorgungs-, als auch für Datenleitungen. Dabei sind einige Parameter wie Clearance und Creepage Distanzen, in Abhängigkeit der auftretenden Spannungen zu beachten. Weiters werden durch die Trennung Störungen durch Masseschleifen und Common-Mode Gleichspannungseffekte reduziert bzw. vermieden.

Zur galvanischen Entkopplung des Messteils vom Mikrocontrollerteiler wird ein 100 Mbaud Digital Isolator HCPL-091J mit einer kurzen Verzögerungszeit von maximal 15

ns und einer Isolationsspannung bis zu 2500 Vrms verwendet. Er stellt somit ein sicheres Interface zwischen der Hochspannungs- und der Niederspannungsseite dar. Der verwendete Digital Isolator ist einer gewöhnlichen Optokopplerlösung bei Performance, Stabilität und Leistungsaufnahme deutlich überlegen, da weder Treiber noch andere externe Bauteile bis auf zwei 47nF Kondensatoren, um die Spannungsversorgung an beiden Seiten zu stabilisieren, benötigt werden und somit Optokoppler-typische Designschwierigkeiten, wie auch nicht lineare Übertragungsfunktionen, Temperatur- und Lebensdauereffekte entfallen. Diese Bauserie ist in einem narrow Body mit ungefähr 3.9 mm Breite (Kriechstrecke) verpackt und verfügt über vier Koppler, jeweils zwei in eine Richtung. Bei Datenraten in der Höhe von 10 Mbaud oder darüber ist die Verwendung von Masseflächen für GND1 und GND2 stark zu empfehlen.

Der Grund warum dieser 100 Mbaud Koppler zum Trennen des SPI Bus, dessen maximale Übertragungsgeschwindigkeit in diesem Fall 10 Mbaud beträgt, Verwendung findet ist seine kurze Signalverzögerungszeit, die bei einem synchronen Bus von großer Bedeutung für die korrekte Funktion ist. Wenn nämlich synchrone Signale mit einer Baudrate von 10 Mbaud durch zu große Verzögerungszeiten im Bereich von 25 ns verschoben werden, werden die Bits der Datenleitungen die verzögert beim Mikrocontroller ankommen falsch übernommen. Abbildung 6.4 veranschaulicht die Problematik.

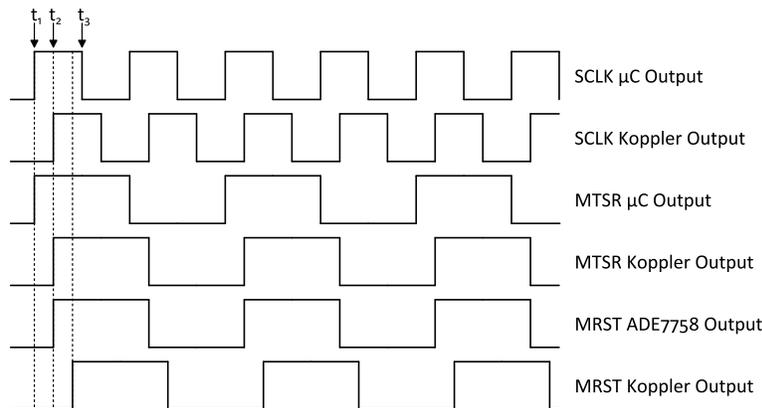


Abbildung 6.4: Verzögerungen am SPI-Bus

- t_1 ... Shiftzeitpunkt des μC
- t_2 ... Verzögerung nach Koppler
- t_3 ... Zeitpunkt der Datenübernahme an MRST durch μC

Der Mikrocontroller erhält die Daten am MRST Eingang um die doppelte Verzögerungszeit verschoben, übernimmt sie aber synchron zu seiner nicht verzögerten ausgegebenen Clock. Der Slave hat kein Problem mit dieser Verzögerung die richtigen Daten zu übernehmen, da sowohl Clock als auch die Signale auf der MTSR Leitung um die gleiche Zeitspanne verschoben sind. Er gibt nur die Daten an seiner MRST Leitung um diese Verzögerungszeit verspätet aus. Diese Zeitspanne der Verzögerung wird dann durch den Koppler noch verdoppelt, bevor das Signal am Master anliegt und die Bits dem Signalpegel entsprechend übernommen werden.

Wie die Abbildung 6.4 verdeutlicht, werden zum Zeitpunkt t_3 die Daten an der MRST Leitung durch den Mikrocontroller festgehalten, was in diesem Fall schon fast kritisch ist, da durch ein wenig Verzögerung mehr, die Daten falsch erfasst würden. Die maximale Verzögerungszeit von 15 ns (typisch 10 ns) des HCPL -091J führt zu einer Gesamtverzögerung der Daten an MRST von maximal 30 ns, was bei einer Clock High-Dauer von 50 ns zu einer beruhigenden Zeitreserve führt.

Eine andere Möglichkeit dieses Problem mit der Verzögerung zu umgehen wäre die Verwendung zweier SPI Schnittstellen des Mikrocontrollers sofern vorhanden (mit XC167 möglich). Hierzu wäre die eine Schnittstelle als Master (SSC0), die andere als Slave (SSC1) zu definieren und der ADE7758 mit der SSC0 MTSR Leitung über den Koppler mit DIN, und DOUT mit MTSR der SSC1 zu verbinden. Die Masterclock der SSC0 würde über den Koppler dem ADE7758 den Takt zur Verfügung stellen und gleichzeitig nochmals über den Koppler laufend, den für den Slave SSC1. Die nicht verwendeten Leitungen wären offen oder auf Massenpotenzial zu legen. Nach dieser Spezifikation dient die SSC0 Schnittstelle ausschließlich zum Senden und die SSC1 Schnittstelle, an der die doppelt verzögerten Daten mitsamt der doppelt verzögerten Clock anliegen, auf deren Basis in der Slave Konfiguration Bits festgehalten werden, zum Empfangen.

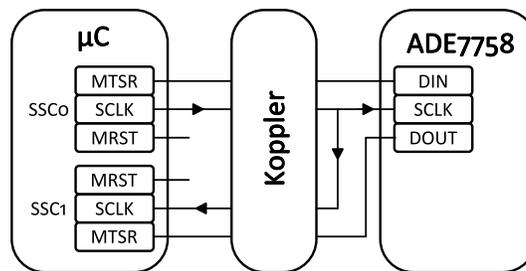


Abbildung 6.5: Verwendung von zwei SPI-Schnittstellen

Diese Lösung wäre sicherlich die kostengünstigere, da es nicht sehr viele verfügbare Koppler mit der nötigen kurzen Verzögerungszeit gibt, aber der entscheidende Nachteil liegt in der Tatsache, dass die zweite SPI-Schnittstelle nicht mehr für anderweitige Kommunikation genutzt werden könnte.

6.3 Schaltungslayout

6.3.1 PCB

Die Platine sollte im Idealfall so aufgebaut sein, dass weder elektromagnetische Störungen oder elektrostatische Entladungen die Schaltung beschädigen, noch permanent beeinflussen können. Da die Signale, die an den analogen Eingängen des ADE7758 anliegen, mit einem Wertebereich zwischen $\pm 0.5\hat{V}$ relativ klein sind, ist eine gewisse EMV technische Beachtung sinnvoll, um die Genauigkeit und Funktion des Messgeräts durch die Verhinderung von Störeinkopplungen sicherzustellen. Wobei nicht immer separate Lösungen gefunden werden müssen, da die RC-Tiefpass Aliasing Filter an den Analogeingängen auch einen EMV Schutz darstellen. Durch ein erweitertes Eingangsschaltungsnetzwerk sollten Störungseinkopplungen über die Messleitungen zusätzlich minimiert werden.

Die Masseflächen wurden so groß wie möglich gemacht, um die gemeinsame Erdimpedanz niedrig zu halten. Außerdem reduzieren sie Interferenzen, welche durch Strahlung aufgenommen werden, sowie verhindert das Risiko von unerwünschten Masseschleifen durch Mehrfacherdung. Die ruhige Analogmassenfläche und die störungsbehaftete Digitalmassenfläche sind physikalisch getrennt und miteinander nur an einem Punkt (Star ground configuration) verbunden. Die Verbindung erfolgt über einen Ferrit, um die Störungen zu minimieren, die von der digitalen Massenfläche auf die analoge Massenfläche eingehen. In einer Applikation-Note von Analog Device wird die Verbindung mittels eines Ferrits empfohlen, doch sei hier vermerkt, dass Ausgleichströme zwischen den Masseflächen aufgrund der für hohe Frequenzen hochohmigen Verbindung dann über den Energy Metering IC selbst fließen können.

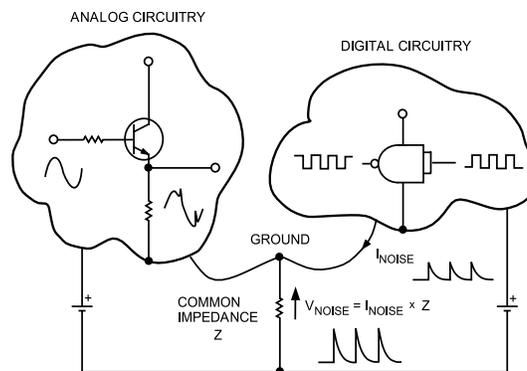


Abbildung 6.6: Erdimpedanz[Col].

Die Signalfade der Strom- und Spannungsleitung zu den Analogeingängen sind möglichst kurz zu gestalten und die Filterschaltungen nahe den Eingängen des Energy Measurement IC zu positionieren, damit die geschützten Leitungen so kurz wie möglich ausfallen.

Leitungen mit schwachen differenziellen Signalen, wie die des Eingangsnetzwerkes zur Strommessung, sind parallel und nahe aneinandergelegt ausgeführt und können noch zusätzlich durch die Massefläche geschirmt werden.

6.3.2 Allgemeine Sicherheit, Kriech- und Luftstrecken

Der Messteil oder Hochspannungsteil, der auf dem Potenzial des Bezugsleiters der Spannungsmessung liegt, ist vom Mikrokontrollerteil oder Niederspannungsteil zu entkoppeln bzw. zu isolieren. Bewerkstelligt wird dies wie bereits erwähnt mit einem DC/DC-Wandler an den Versorgungsleitungen und dem Digital Isolator an den Datenleitungen. Da keine gefährlichen Spannungen an der sicheren Schaltungsseite des Mikrocontrollers oder dem Gehäuse auftreten sollen, gilt es beim Entwurf besonders auf mögliche Fehlerfälle wie zum Beispiel, Kurzschlüsse oder Überspannungen im Messteil, und die daraus abgeleiteten Regeln in Bezug auf Sicherheit und Isolierung zu achten.

Grundsätzlich gilt, dass zwei Levels an Sicherheit zwischen aktiven und sicheren Teilen wie dem Gehäuse sein sollten. Das kann durch einen Layer Isolation zuzüglich einer Erdung des metallischen Gehäuses (Geräteklasse 1) oder durch zwei Layer an Isolation und dem Schutz vor direktem Berühren durch ein Kunststoffgehäuse (Geräteklasse 2) erreicht werden. Der erste Layer an Isolation wird als Basic Insulation bezeichnet. Basis Isolierung

muss zwischen aktiven Teilen und Teilen, die geerdet sind gegeben sein. In der vorliegenden Schaltung bezieht sich dies auf die Eingangsstruktur des Messteils, an der Netzspannung anliegt, und die geerdeten Teile bzw. dem damit verbundenen Gehäuse. Für die Basis Isolation sollten deren Abstände an Luft- und Kriechstrecke nicht weniger als 4mm betragen und die Platine nach Möglichkeit auch noch mit einer Isolierschicht überzogen werden, damit in einer rauen Umgebung die Isolierfestigkeit nicht durch Faktoren wie Schmutz, Korrosion oder Luftfeuchtigkeit sinkt.

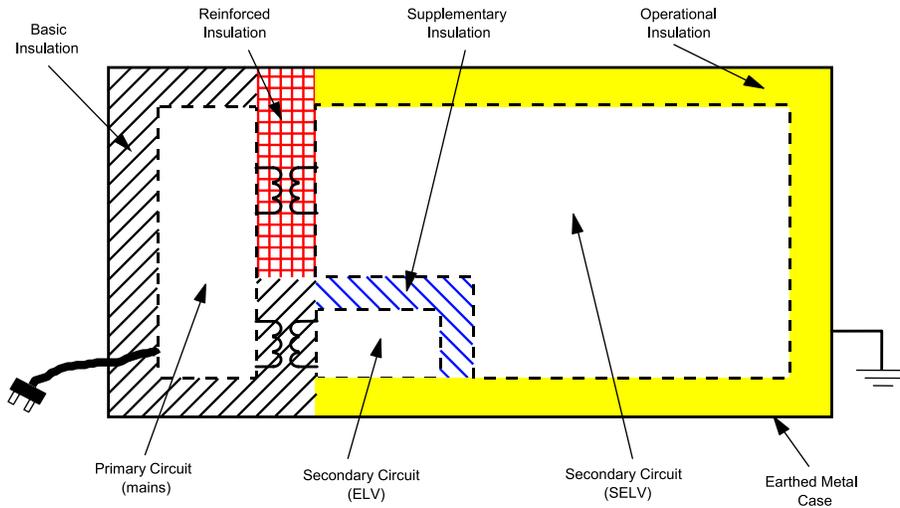


Abbildung 6.7: Klassen von Isolierung[Law].

Wenn keine Erdung vorhanden ist, wird ein zweiter Layer an Isolierung nötig, der als Supplementary Insulation bezeichnet wird. Die Kombination dieses Layers mit dem Layer der Basis Isolation ergibt die Double Insulation. Oft gibt es einen einzelnen Layer an Isolation zwischen aktiven und sicheren Teilen, der dieselben Isolierungseigenschaften wie die Double Insulation aufweist, und als Reinforced Insulation bezeichnet wird (vgl. Abb. 6.7). Die aus Sicherheitsgründen nötigen Abstände zur Isolierung zwischen primären Schaltungsteilen mit hohen, im Betrieb und Fehlerfall möglichen Spannungen und sekundären, sicheren Schaltungsteilen, sowie der unmittelbaren Umgebung wie dem Gehäuse werden auch als Creepage- und Clearance-Distanzen bezeichnet.

Creepage Distance bezeichnet den kürzesten Weg zwischen zwei leitenden Teilen (oder zwischen einem leitenden Teil und einer Grenzfläche des Gerätes), gemessen entlang der Fläche der Isolierung. Ausreichende Kriechdistanzen schützen gegen Tracking, welches einen partiellen leitenden Pfad aufgrund von lokalen Verschlechterungen der Isolation auf der Fläche hervorruft, als Resultat elektrischer Entladungen an oder in der Nähe der isolierenden Fläche. Es tritt häufig in rauerer Umgebung auf. Clearance Distance bezeichnet die kürzeste Distanz zwischen zwei leitenden Teilen (oder zwischen einem leitenden Teil und einer Grenzfläche des Gerätes), gemessen an der Luftlinie. Die Distanz verhindert dielektrische Zusammenbrüche zwischen Elektroden aufgrund von Ionisierung der Luft (vgl. Abb. 6.8). Die Kriech- und Luftstrecke zwischen primären und sekundären Schaltkreisen, die nicht geerdet sind, sollte für Reinforced Insulation mindestens 8mm betragen. (vgl. [Hom03])

Die geeigneten Creepage- und Clearance-Distanzen können aus den Tabellen in der

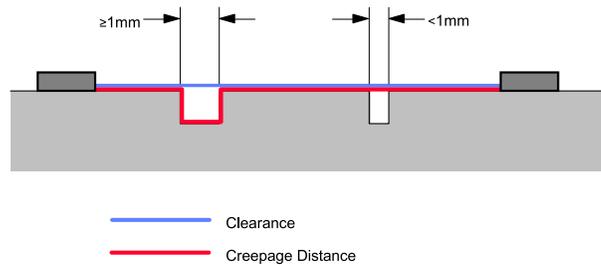


Abbildung 6.8: Gegenüberstellung Kriech- und Luftstrecke[Law].

Norm EN 60950 entnommen bzw. daraus berechnet werden (siehe Anhang). Folgende Faktoren spielen dabei eine entscheidende Rolle und müssen berücksichtigt werden: Betriebsspannung, Verschmutzungsgrad der Umgebung und auch die Überspannungskategorie bzw. für die Kriechstrecke ist auch die Materialgruppe, nach CTI (Comparative Tracking Index) Performance Grade spezifiziert, mitbedeutend.

Zum Schutz vor Überspannungen und energiereichen niederfrequenten Störungen kann das Eingangsspannungsnetzwerk an den Leitern noch mit Varistoren auf den Neutralleiter beschalten werden. Die eingesetzten Varistoren müssen dabei einen Schwellspannungswert aufweisen, der groß genug dimensioniert ist, da Varistoren durch die ankommenden Überspannungen Alterungsprozessen unterworfen sind und dabei niederohmiger werden, wodurch sich Leckströme ausbilden können. Der aufgedruckte Wert entspricht dem Effektivwert der maximal zulässigen Wechsellspannung, die dauerhaft anliegen darf. Geht man unter der allgemeinen Dimensionierungsvorschrift für Varistoren, mit 10-20% Spannungstoleranz zur Nennbetriebsspannung, davon aus, dass das Netz immer wieder Schwankungen unterworfen ist, und bei der Messungen an Lasten in Dreieckschaltung 400V Außenleiter-Spannung am Messgerät liegen, bietet sich beispielsweise ein Wert von 460V (S10K460) an. So schützt der Varistor auch die Widerstände des Spannungsteilers vor dem Erreichen der für sie ausgewiesenen Grenzspannung.

Zur Unterdrückung von hochfrequenten Störeffekten, die über die Messspannungsleitungen eingekoppelt werden und das Gerät in ihrer Funktion beeinträchtigen und stören können, ist es möglich Y-Kondensatoren zwischen Leitern, Neutralleiter und Schutzleiter bzw. Gehäuseerde zu schalten. Diese zeichnen sich durch besondere Sicherheit sowie Selbstheilungsfähigkeit aus und dämpfen Gleichtakt-Störimpulse gegen Erde. Der Wert der Kondensatoren (1nF) und der ESL (Ersatz-Serien Induktivität), die im Hochfrequenzbereich eine entscheidende Rolle spielt, sollte niedrig sein, um die Eigenresonanzfrequenz, bei der die Impedanz am geringsten ist, zu möglichst hohen Frequenzen zu bewegen. Werden Y-Kondensatoren bei dem vorliegenden Messgerät eingesetzt, muss auf eine korrekte Erdung des Schutzleiters geachtet werden, da falls der Schutzleiter nicht angeschlossen ist, die Kondensatoren einen kapazitiven Spannungsteiler im dreiphasigem Netz bilden und das Gehäuse, das mit dem Schutzleiteranschluss verbunden ist, somit unter Spannung setzen.

Zusätzlich können zur Störunterdrückung noch Längsinduktivitäten in die Messleitungen der Spannungssignale und des Neutralleiters eingebracht werden. Anzumerken ist, dass diese Induktivitäten den Phasengang, sowie die Amplitude des Signals im beobachteten Frequenzbereich zusätzlich geringfügig beeinflussen können, was aber durch die Kalibrie-

rung berücksichtigt wird.

Da der Schutzleiter bei Geräten der Schutzklasse 1 direkt über eine Schraubverbindung, die keine weitere mechanische Aufgabe erfüllen darf, mit dem Gehäuse verbunden werden muss, wird dieser direkt an diesem Punkt angeschlossen. Somit ist keine groß dimensionierte Leiterbahn auf der Platine nötig, die den Fehlerstrom des Stromkreises tragen muss, da die einzigen Leiterbahnverbindungen mit Erde dann zu den Y-Kondensatoren führen, und somit kleiner ausfallen können. Optional ist auch eine Erdung für den sekundärseitigen Teil der Schaltung vorgesehen, da dieser sonst bei der Verbindung über ein RS232 Kabel mit selbigen am Computer geerdet wird. Das kann zu schädigenden Ausgleichsströmen führen, da das Potenzial der Schaltung floatet, also nicht fixiert ist. Wenn die Bezugsmasse des sekundären Teils der Schaltung und PE miteinander verbunden sind, kann aber zum Nachteil eine Brummschleife entstehen und somit Störungen auftreten, da diese Verbindung im Gesamtsystem, bei angeschlossenem Computer, mehrfach ausgeführt ist.

Das Gehäuse des Prototyps besteht aus eloxierten Aluminiumprofilen. Das Eloxal muss für eine leitende Verbindung an den Verbindungsstellen zum Schutzleiter abgetragen und die Leitfähigkeit überprüft werden. Durch die Profilbauweise sind die einzelnen eloxierten Elemente nicht leitfähig miteinander verbunden. Aus diesem Umstand ist nur die Bodenplatte mit Schutzleiteranschluss direkt geerdet, wobei die Basis Isolierung zu den aktiven Teilen unter der Berücksichtigung eines nötigen Abstandes von 4mm eingehalten werden muss. Zur Einhaltung der sicherheitsrelevanten Bestimmungen muss weiters der Abstand der Teile zu den anderen Gehäuseelementen und Schaltungsteilen dann mindestens 8mm betragen (Reinforced Insulation). Ergeben sich beim Gehäuseeinbau Luftstecken, die kleiner als die geforderte Distanz zur Einhaltung der Sicherheit sind, kann zusätzlich mit Isolierfolie gearbeitet werden, um die gesamte Durchschlagsfestigkeit zusätzlich zu der der Luft zu erhöhen.

Kapitel 7

Software

7.1 Programm Mikrocontroller

7.1.1 Softwarestruktur

Die Software des Gesamtsystems wird in einzelne Programmteile mit bestimmten Teilaufgaben aufgeteilt, die einzeln klassifiziert werden und gewisse Abhängigkeiten zueinander haben können. Prinzipiell kann bei eingebetteten Systemen zwischen Zeitsteuerung, die in einfachsten Fall aus einer Endlosschleife besteht und Ereignissteuerung, die auf externe Ereignisse reagiert unterschieden werden. Praktisch ist eine Kombination am sinnvollsten, da auch die einzelnen Teilaufgaben von ihren Eigenschaften her diesen Ansätzen zugeordnet werden können. Folglich existieren Task mit strenger Zeitbindung, wie zyklische in exakten Zeitabständen ablaufende Tasks, Ereignis-gesteuerte Tasks, die durch ein externes Ereignis gestartet werden und generelle Tasks mit Zeitbindung, die in einer Endlosschleife abgearbeitet werden. Tasks mit strenger zyklischer Zeitbindung werden als Timer ISRs und Ereignis-gesteuerte Tasks als Event ISRs implementiert. Im folgenden Softwaredesignprozess werden die einzelnen Teilaufgaben entsprechend eingeteilt, wobei auch Prioritäten festzulegen sind, da Tasks mit strenger Zeitbindung und Ereignis-gesteuerte Tasks sich auch zugleich um Rechenzeit bemühen und diese in Anspruch nehmen wollen. Das Ziel dieser Ansätze ist es, die Rechenzeit möglichst effizient für die Ausführung des Gesamtprogramms auf die einzelnen Teilaufgaben bzw. Tasks aufzuteilen. Für das Programm eines eingebetteten Systems folgt mit der Kombination von Zeit- und Ereignissteuerung eine Struktur wie Abbildung 7.1 veranschaulicht.

Da eine Programmverzweigung in eine ISR nach der Freigabe des Interrupts jederzeit erfolgen kann, verhält sich das Programm nicht deterministisch. Die Unterbrechung gewisser Programmteile wie z.B. während eines zeitkritischen Kommunikationszyklus muss durch das Sperren der Freigabe dementsprechend verhindert werden. Weiters muss vorher die maximale zu erwartende Häufigkeit der Ereignisse abgeschätzt werden, um die Wahrscheinlichkeit von Überlastung zu minimieren. Den möglichen Interrupts kann gemäß ihrer Wichtigkeit eine Priorität zugeteilt werden, wobei eine ISR mit einer niedrigen Priorität durch einen Interrupt mit höherwertiger Priorität unterbrochen werden kann. Somit sind die einzelnen Aufgaben auch in ihrer Wichtigkeit einzuteilen. Prinzipiell gilt, die ISR selbst so knapp wie möglich zu halten, um nicht weitere Ereignisse und ihre Behandlung aufzuschieben. Da alle Tasks bzw. Module zur Erfüllung der Gesamtaufgabe zusammen-

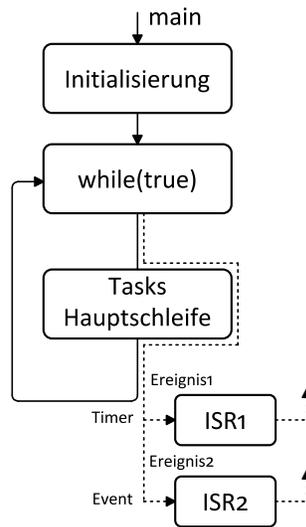


Abbildung 7.1: Softwarestruktur Embedded System

wirken, ist eine geregelte Kommunikation untereinander und eine generelle Abstimmung notwendig. Dies geschieht am Besten über globale Variablen bzw. Semaphoren, die in einem Task bzw. in der ISR gesetzt und von einem anderen Task gelesen und rückgesetzt werden. Flags dienen dabei zur gegenseitigen Abstimmung. Die Kommunikation über Semaphoren sollte nach einem nicht blockierenden Schema von Stattengehen, da Blockaden durch Warten oder Polling Potenzial für Instabilität des Gesamtsystems in sich bergen. Ausgehend von dieser Betrachtung kann die Software eines Embedded Systems anhand seiner zu erfüllenden Teilaufgaben entworfen, modular aufgebaut und in der prozeduralen Programmiersprache C realisiert werden.

7.1.2 Beschreibung und genereller Ablauf

Die Pflicht des Mikrocontrollers ist die Steuerung des Systems sowie die Verarbeitung und Übermittlung der Daten. Der Softwareentwurf zur Durchführung dieser Aufgaben folgt dem Bottom-up Verfahren, bei dem das Programm ausgehend von den kleinsten Unter-einheiten hin zu einem Gesamtsystem zusammenwächst. In diesem Zusammenhang soll hier auf einzelne Teilaufgaben, ausgehend von den Hauptaufgabengebieten und ihren Besonderheiten eingegangen werden. Die Implementierung folgt dem bereits beschriebenen Prinzip der Softwarestruktur für Embedded Systems. Die vereinfachte Form des Flussdiagramms Abb. 7.2 mag einen groben Eindruck über das Programm und dessen Struktur anhand der bedeutungsvollsten Aufgaben geben.

Menü und Programmfluss

Der Programmfluss wird durch den Mikrocontroller sowohl mit Zeitsteuerung, bei zyklisch geforderten Abläufen, als auch mit Ereignissteuerung, wenn auf ein von außen kommendes Ereignis mittels Interrupt reagiert werden muss, gelenkt. Der Programmablauf und so die allgemeine Funktionalität können durch den Benutzer über das Tastenfeld im Zusammenhang mit den am Display gebotenen Auswahlmöglichkeiten eines Menüs gesteuert werden.

Da dabei die Eingangsgrößen in Abhängigkeit von dem jeweiligen Zustand auf den Programmablauf Einfluss nehmen, empfiehlt sich ein Entwurf, der in seiner strukturierbaren Form dem einer statusorientierten Maschine entspricht. Dabei folgt auf eine bestimmte Eingabe eine vom momentanen Zustand abhängige Aktion sowie gegebenenfalls auch ein Wechsel des Zustands. Ein State Diagramm, welches diesen Ablauf genau beschreibt, kann auch anhand zweier Tabellen, die in Abhängigkeit von der Eingangsgröße und dem Zustand einerseits die Aktion und andererseits den nächsten Zustand bestimmen dargestellt werden. Die Implementierung in Code erfolgt mittels Arrays, die dementsprechend über die Eingangsgrößen und die Zustände indiziert werden. Dabei wird bei einer Eingabe zuerst über einen Pointer indirekt die Funktion aufgerufen, die für diese Eingabe und den momentanen Zustand definiert ist und anschließend aus dem zweiten Array der Folgezustand ermittelt.

Anhand dieser Methode lassen sich eine Menüführung und damit verbundene Aktionen realisieren. In vorliegender Applikation setzt ein Tastendruck, der eine bestimmte Eingangsgröße repräsentiert und über einen Interrupt detektiert wird, diese statusorientierte Maschine durch einen Funktionsaufruf, der abhängig vom momentanen Zustand ist in Gang. Die Zustände dieser Maschine spiegeln sich gewissermaßen in den Auswahlmöglichkeiten am Display wider. Bei einer Zustandsänderung der Maschine, bedingt durch einen Tastendruck, ändert sich gleichermaßen auch die Displayausgabe, da ein neuer Zustand in Kombination mit der Anzeige am Display und einem Tastendruck auch eine andere auszuführende Funktion repräsentiert. Bei der Anzeige von Messwerten werden diese mit der Frequenz des entsprechenden zyklischen Tasks aktualisiert.

Streng zyklische Abläufe und Tasks werden durch einen Timer gestartet, der beim Erreichen seines Höchststandes einen Interrupt erzeugt, welcher das Programm zu einer Verzweigung in die ISR veranlasst. Die Periodendauer des Auftretens dieses Interrupts muss dabei groß im Vergleich zur abschätzbaren Ausführungsdauer der Tasks sein, um einen flüssigen Ablauf zu gewähren und allen auftretenden Ereignissen schnellstmöglich nachzukommen. Da bei großen Periodendauern die Ausführungszeiten nicht weiter ins Gewicht fallen und falls die Frequenz der auftreten Tasks um diese Ausführungszeiten schwanken darf, kann ein Timer zur Bereitstellung mehrerer Zeitintervalle, die alle ein Vielfaches vom Interruptintervall sind, genügen. Dazu wird eine Variable im Takt des Timer-Interrupts bis zum gewünschten Vielfachen dieses Taktes inkrementiert und stellt somit abhängig von ihrem Wert gewissermaßen einen eigenen Zeittakt dar.

Aufgaben in der Hauptschleife fallen in die Kategorie der generellen Tasks mit Zeitbindung. Dabei ist nicht so wichtig wann genau, sondern das die Aufgaben in definierter Reihenfolge immerwährend bewältigt werden. Dieser Umstand erlaubt es Bedingungen, deren Zustände sich im Laufe des gesamten Programms ändern können, zu prüfen und daran geknüpfte Aktionen auszuführen sowie Ereignisse zu erfassen, die sich nicht durch einen Interrupt kundtun, sondern einer kontinuierlichen Abfrage bedürfen.

Messung und Datenverarbeitung

Das Programmteil der zuständig für den Ablauf der Messung und die Erfassung der Messdaten ist, ist Ereignis-gesteuert aufgebaut, da der Energy Metering IC alle, im Zusammenhang mit der Messung relevanten Ereignisse über einen Interrupt signalisiert. So kann die Zeit zwischen den Interrupts für andere Anwendungen zur Verfügung stehen, wo-

bei immer der Interrupt ausgelöst durch den ADE7758 höchste Priorität genießt und nicht etwa durch einen Anderen unterbrochen wird. Nach dem Setzen der Interrupt-Mask Bits, von denen jedes für sich genommen beim Auftreten eines bestimmten Ereignisses einen Interrupt ermöglicht, wird auf eine fallende Flanke am Interrupt Pin gewartet, welche durch eine Übereinstimmung eines Bits des Status-Registers mit dem Interrupt-Mask-Register ausgelöst wird.

Mit dem Freigeben der einzelnen Interrupt-Quellen muss achtsam umgegangen werden, da unter Umständen zu viel Rechenzeit in Anspruch genommen wird, wenn zu viele Interrupts ermöglicht werden, von denen bereits absehbar ist, dass sie in hoher Frequenz auftreten und der Mikrocontroller öfter als eigentlich nötig in die ISR verzweigen, und damit verbundene Aktionen abarbeiten muss. Falls ein Interrupt periodisch auftritt, aber die damit verbundene Konsequenz in der bestehenden Frequenz unnötig ist, kann ähnlich einer festgelegten Zeitspanne in einem Scheduler, die Freigabe dieser Interrupt Quellen, durch Setzen und Rücksetzen des entsprechenden Bits im Interrupt-Mask-Register an ein zyklisches, vorhersehbares Ereignis gekoppelt werden. In bekannter Häufigkeit wird der Interrupt somit ermöglicht und nach dem Auftreten von selbigen für eine gewisse Zeitdauer wieder gesperrt.

Sollen die Effektivwertregister von Strom und Spannung immer nur in Verbindung mit einem periodisch auftretenden Nulldurchgangs-Interrupt der entsprechenden Phase gelesen werden, um die Auswirkungen des am Signal vorhandenen Rippel zu minimieren, tritt im dreiphasigen System dieser Interrupt häufiger als nötig auf, um den eigentlichen Bedarf an aktuellen Effektivwerten zu decken. Durch die Kopplung der Freigabe der Interruptquelle an ein anderes zyklisches Ereignis kann die Auftrittshäufigkeit auf ein zweckmäßiges Maß gesenkt werden.

Der aktuelle Wert des 24 Bit breiten Datenwortes des Mask-Registers, welches alle möglichen Interrupt-auslösende Quellen umfasst, muss dem Mikrocontroller bekannt sein. So ist es diesem möglich durch eine Verknüpfung einen neuen Sollwert für dieses Register zu formen, falls bestimmte Bits unabhängig vom Rest verändert werden sollen. Dazu wird eine globale volatile long Variable deklariert, die den letzten geschriebenen und somit derzeitigen Wert des Interrupt-Mask-Registers beinhaltet und immer aktualisiert wird, sobald ein neuer Wert in Mask-Register geladen wird. Diese Variable dient auch als Maske, um in der ISR zu evaluieren was für ein Art Interrupt aufgetreten ist. Der 24 Bit umfassende Wert des Status-Registers wird nun verwendet, um die Quelle bzw. die Art des Interrupts zu bestimmen, indem er mit der globalen Variablen, die den aktuellen Wert des Mask-Registers beinhaltet, logisch verglichen wird. Daraus ergibt sich die Interrupt-auslösende Quelle, woraufhin der Mikrocontroller eine entsprechende Aktion bzw. einen Datentransfer durchführen kann.

Nach Setzen des Interrupt-Mask-Registers und der Übereinstimmung eines Flags mit dem Status-Register geht der \overline{IRQ} Ausgang sofort auf Low-Pegel. Dieser Umstand kann zu Missverständnissen und damit Fehlinterpretationen führen, wenn das entsprechende Flag des Status-Registers zu diesem Zeitpunkt bereits gesetzt war und die fallende Flanke von \overline{IRQ} somit nicht mit dem Auftreten des eigentlichen Ereignisses zusammenfällt. Das kann insofern kritisch sein, wenn bei der Art des Interrupts der Faktor Zeit zwischen dem tatsächlichen Auftreten und der damit verbundenen, sofort folgenden Aktion eine Rolle spielt. Dies zeigt sich beim kontinuierlichen Setzen und Rücksetzen des Interrupt-Mask Bits für die Interruptauslösung bei Nulldurchgängen der einzelnen Phasen. In solchen

Fällen gilt es, die geplante Aktion sofern zulässig einfach einmalig auszusetzen bzw. den ersten Interrupt zu ignorieren und auf den Nächsten dieser Art zu warten. Dadurch ist sichergestellt das der Interrupt, ausgelöst durch die fallende Flanke mit dem Zeitpunkt des korrespondierenden Ereignisses übereinstimmt.

Die Justierung und die Datenverarbeitung der vom ADE7758 gewonnenen diskreten Messwerte erfolgen auf Basis der bei der Kalibrierung ermittelten Parameter. Bei den errechneten Koeffizienten zur Korrektur oder Anpassung der Messwerte im Energy Measurement IC selbst, handelt es sich um statische Werte, die nach jedem Einschaltvorgang bzw. jeder Unterbrechung der Versorgungsspannung in die dafür vorgesehenen flüchtigen Register geladen werden müssen. Diese nehmen eine grobe Anpassung der einflussreichsten Fehler bei der Berechnung der Messwerte aus den aufgenommenen Werten vor. Werden die Werte durch Division angepasst, ist darauf zu achten, diesen Prozess durch Multiplikation im Mikrocontroller rückgängig zu machen, noch bevor mit den Daten anderweitige Operationen ausgeführt werden. Im Mikrocontroller bieten sich zusätzliche Möglichkeiten zur Justierung der ermittelten Werte und Korrektur der Umsetzerkennlinie an. Durch die Aufnahme vieler Referenzpunkte bei der Kalibrierung lassen sich vorhandene Nichtlinearitäten berücksichtigen und herausrechnen. Auch ist es möglich mit intervallweiser Approximation für gewisse Messbereiche bestmögliche Koeffizienten zu errechnen, die dann für weitere Berechnungen herangezogen werden. Anschließend wird zu dem korrigierten Wert ein entsprechender physikalischer Messwert ermittelt.

Datenübertragung

Eine weitere Aufgabe des Mikrocontrollers als zentraler Knoten und Umschlagpunkt ist die Datenübermittlung zwischen den einzelnen Schnittstellen und den daran befindlichen Teilnehmern nach gegebenenfalls zwischenzeitiger Aufbereitung dieser Daten.

Der Datenaustausch zwischen der Messeinheit ADE7758 und der Verarbeitungseinheit Mikrocontroller umfasst sowohl Kontroll-Daten, die der Mikrocontroller übermittelt und zur Steuerung des Mess-IC dienen, als auch Mess-Daten, die der ADE7758 auf entsprechende Anfrage ausgibt. Die Grundlage der Kommunikation mittels der SPI Schnittstelle bildet dabei eine Routine zum Senden bzw. zum Empfangen von einzelnen Bytes. Zuerst wird der Transmit-Buffer geladen und der Datenaustausch durch Polling des receive Interrupt request Bits (SSCRIR), welches anzeigt, dass die Übertragung zu Ende ist, gesteuert. Anschließend kann aus dem Receive-Buffer das empfangene Byte gelesen werden. Da der Datenaustausch synchron an beiden Datenleitungen passiert, wird beim Schreiben ein Byte mit keinem Informationsgehalt empfangen und beim Lesen selbiges geschrieben. Wichtig beim Polling ist das manuelle Rücksetzen des Interrupt Receive Bits SSCRIR.

Aus diesen Grundroutinen werden Funktionen zum Lesen und Schreiben mehrerer Bytes aufgebaut, die der zeitlichen Abfolgen entsprechen. Da die Registergröße unterschiedlich ist, muss sie diesen Funktionen übergeben und falls nötig, auf die nächste gerade Anzahl von Bytes gerundet werden. Bei einer Baudrate von 10 Mbaud und aufgrund der schnellen Ausführung von Code durch den XC167 muss besonders auf die Verzögerungszeit zwischen dem Schreiben der Commandbytes und dem nachfolgenden Lesevorgang der Register (min. 4s) geachtet werden. Aus diesem Grund muss hier auch gegebenenfalls eine explizite Verzögerung eingebaut werden.

Um sicherzugehen, dass kein Interrupt bzw. eine Programmverzweigung während dieser

Kommunikationsphase auftritt, die die Datenübertragung unterbricht, werden die Interrupts für diese Zeit gesperrt. Die aktuell freigegebenen möglichen Interrupts sollten bei Funktionseintritt zwischengespeichert werden, sodass bei Funktionsende der vorherige Zustand wiederhergestellt werden kann und es nicht zu einer unabsichtlichen Aktivierung eines möglichen Interrupts und damit zu einem nicht vorhersehbaren und unbeabsichtigten Ablauf kommt.

Messwerte werden in einem wählbaren Intervall in einen char String umgewandelt und als solcher in eine Software Buffer geschrieben bevor sie an den USB Host-Controller bzw. den USB Flash Speicher übertragen werden. Die Daten werden dabei nach der Struktur einer tabellenartig organisierten CSV-Datei abgelegt, sodass sie direkt in eine solche Datei überführt werden können. Die Standard Sektor Größe von USB Flash Speichern ist 512 Byte, wobei die beste Performance erreicht wird, wenn immer Blöcke dieser Größe geschrieben werden. Außerdem lässt sich durch die Zwischenspeicherung vermeiden, dass der Mikrocontroller permanent mit der Datenübertragung beschäftigt ist. Diese Zwischenspeicherung erfolgt in zwei 512 Byte großen Buffern, die im Speicher nebeneinanderliegen, sodass Daten in den zweiten Buffer zwischengespeichert werden können, während aus dem ersten vollen Buffer Daten an den USB-Controller übertragen werden. Eine Grundroutine, die an den speziellen Aufbau der Schreib- und Lesezyklen des USB Host-Controllers angepasst ist, managt den Datenaustausch auf unterster Ebene. Die Kommunikation zur Steuerung erfolgt über Kommandos, die von einem Monitor interpretiert und ausgeführt werden. Da dieser die Kommandos nicht nur quittiert, sondern auch gegebenenfalls selbstständig Rückmeldungen liefert, muss das Mikrocontrollerprogramm flexibel auf diese Antwort reagieren und darauf warten, bevor ein weiteres Kommando erteilt wird, da es sonst zu Fehlerzuständen kommen kann. Die für die Speicherung am USB Flash-Memory nötigen Kommandos sind im Programm als Funktionen definiert. Etwaige Meldungen, die der USB-Controller generiert, und die auf ein eingetretenes Ereignis hinweisen, wie das Anstecken oder Abziehen eines Flash Speicher Sticks müssen durch den Mikrocontroller selbst kontinuierlich abgefragt werden. Der USB-Controller managt das FAT File System von sich aus und legt die Daten in der dafür geöffneten *.csv Datei der Reihe nach ab. Nach Beendigung der Aufzeichnung wird die Datei mit Komplementierung der FAT geschlossen und ist als solche am Computer ersichtlich.

Der Datenaustausch zwischen Mikrocontroller und XPort geschieht asynchron über die serielle Schnittstelle. Den eigentlichen Kommunikationsendpunkt stellt aber das Java Applet dar. Bei der Verschiedenartigkeit der übertragenden Information bedarf es eines für die Anwendung spezifischen Protokolls, welches auch an die beiden Kommunikationsparteien angepasst ist. Dies verlangt eine mit dem Java Applet zusammenhängende Betrachtung.

Speicherung internes Flash Memory

Um nicht durch eine Unterbrechung der Spannungsversorgung bzw. einen Netzausfall momentane Programmzustände oder Energiewerte, die bereits stetig über einen Zeitraum angesammelt wurden, zu verlieren, müssen diese nicht flüchtig im Flash abgespeichert und beim erneuten Einschalten wieder geladen werden. Die Speicherung kann kontinuierlich geschehen und/oder über einen Interrupt, der durch eine SAG Dedection des ADE7758 ausgelöst wird und einen Spannungsabfall des Netzes unter einen bestimmten Wert anzeigt. Wenn auch die gesamte Messschaltung vom selbigen Netz aus versorgt wird, bedeu-

tet dies, dass die Versorgungsspannung kurz vor einer möglichen Unterbrechung steht. Ist der Mikrocontroller rechtzeitig gewarnt, kann noch eine rettend Speicherung ausgeführt werden.

Programm und Daten werden im gleichen Flashspeicher abgelegt. Da aber während der Flash Programmierung kein Code vom Speicher, der gerade programmiert wird ausgeführt werden kann, muss das Programm während des Schreibens der Daten aus dem internen RAM exekutiert werden. Dazu müssen die Flash Programmier Routinen ins RAM geladen und von dort ausgeführt werden. Der L166 Linker/Locator ermöglicht es Programmcode im Flash-Memory zu speichern und eine andere Exekutionsadresse (z. B. im RAM) dafür zu definieren. Ein zusätzliches Header File muss inkludiert werden, das Makros beinhaltet welche die ROM Adresse, die RAM Adresse und die Länge der Flash Programmerroutinen bereitstellt. Dies ist nötig, da sowohl die Startadresse als auch die Länge der Routinen bekannt sein müssen, um diese vor ihrer Ausführung ins RAM umzusiedeln. (vgl.[Kei99])

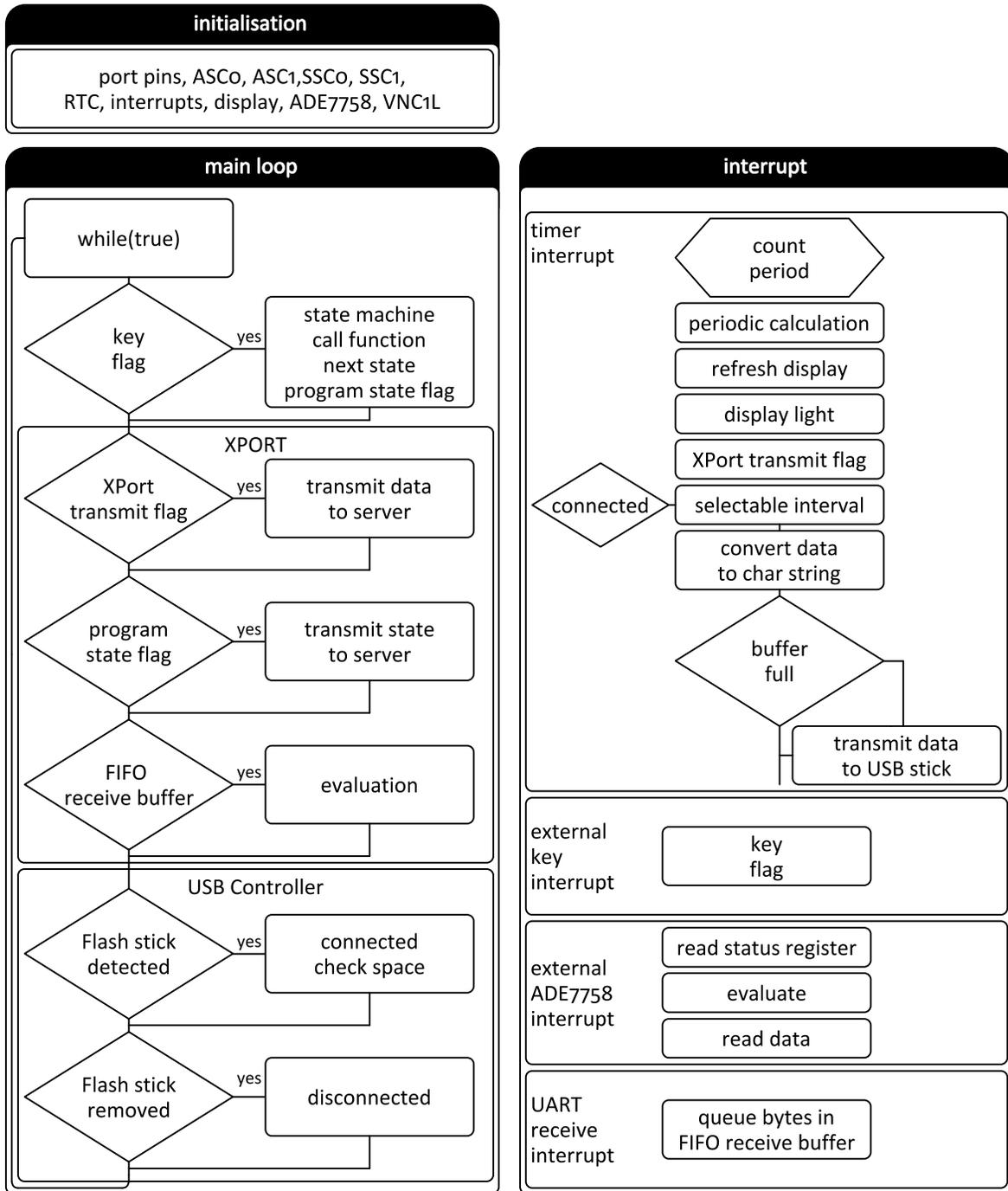


Abbildung 7.2: Flussdiagramm des Programms

7.2 Java Applet

Ein Java Applet ist ein Java Programm, dass von einem Server, durch den Webbrowser eines Clients heruntergeladen und in diesem dann, in der Java Virtual Machine innerhalb einer Sandbox, ausgeführt wird. Das Applet ist dabei üblicherweise in eine HTML Seite eingepackt (vgl. Abb. 7.3).

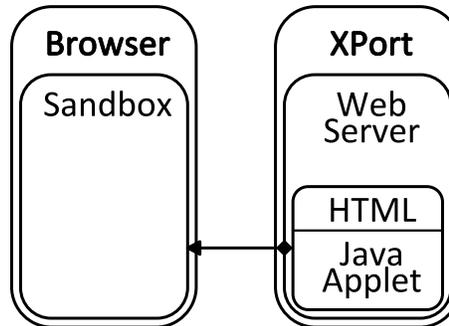


Abbildung 7.3: Ausführung Java Applet

Wird eine Seite mit Java Applet durch den Browser geladen, beginnt dessen Lebenszyklus, auf Basis von drei Zuständen und vier Methoden (vgl. Abb. 7.4).

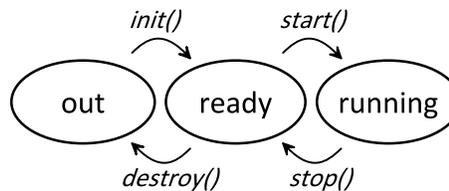


Abbildung 7.4: Lebenszyklus des Java Applet

Nach dem Laden ruft der Browser die *init()* Methode auf, die ähnlich einem Konstruktor die Attribute initialisiert.

Unmittelbar nach der Initialisierung wird mit der *start()* Methode der eigentliche Programmablauf des Java Applet gestartet. Der Aufruf dieser Methode erfolgt auch jedes Mal wenn die HTML Seite neu geladen wird oder der Benutzer nach Aufruf anderer Seiten wieder zu dieser Seite zurückkehrt.

Wird die Webseite, auf der das Applet eingebunden ist, verlassen, erfolgt der Aufruf der *stop()* Methode, aber auch wenn das Browser Fenster überdeckt und inaktiv wird.

Die *destroy()* Methode löscht das Applet Objekt aus dem Hauptspeicher. Zu beachten gilt, dass die *destroy()* Methode nur aufgerufen werden kann, wenn der Browser ordnungsgemäß beendet wird.

Jedes Applet muss eine Subklasse der Applet Klasse sein, das heißt von ihr erben, da diese Klasse das Interface zwischen Applet und der Browser Umgebung darstellt. Applets unterliegen aufgrund der Tatsache, dass sie potenzielle schädliche Programme anderer Rechner

sind, die aus dem Netz heruntergeladen werden, bestimmten Sicherheitsbestimmungen, die den allgemeinen Funktionsumfang des Java Programms einschränken können. Dazu gehören:

- Kein Zugriff auf das Dateisystem
- Kein Schalten von Netzwerkverbindungen
- Kein Starten anderer Programme
- Keine Druckmöglichkeit

Es ist auch möglich ein Java Programm so anzulegen, dass dieses sowohl als Applet fungiert und von einem Server geladen werden kann, beziehungsweise als eigentliches Java Programm arbeitet, das lokal auf einem Rechner abgespeichert ist und auf diesem ausgeführt wird.

7.2.1 Socket

Ein Socket ist eine standardisierte, plattformunabhängige Schnittstelle, die ein Programm mit einem Netzwerk verbindet und dabei einen Kommunikationsendpunkt darstellt. Ein Internet Socket lässt sich durch das verwendete Transport Protokoll, die IP-Adresse und die Portnummer beschreiben. Diese Information bildet die Socket-Adresse. Durch Extraktion dieser Adresse aus dem Protokoll Header ist der Zielpunkt in Form einer entsprechenden Anwendung, zu der der eingehende Netzwerkverkehr weitergeleitet wird, definiert.

Über ein eigenes Socket kann ein Java Applet einen Kommunikationskanal ausschließlich zu dem Host von den es heruntergeladen wurde öffnen. Im Falle des XPORT, das alle Daten die über die serielle Schnittstelle einlangen an den TCP Port 10001 und umgekehrt weitergibt, resultiert für den Kommunikationsablauf eine Topologie nach Abbildung 7.5.

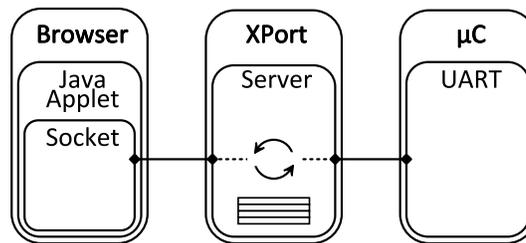


Abbildung 7.5: Kommunikationskanal Applet - XPort - Mikrocontroller

7.2.2 MVC

Im Sinne einer Objekt-orientierten Programmierweise wird eine Anwendung in mehrere Klassen aufgeteilt. Ein Modell für diese Aufteilung und die Strukturierung liefert das MVC Konzept (Model, View, Controller) mit dem Ziel eines flexiblen Progammentwurfes. Dabei wird die Anwendung anhand von einzelnen Klassen oder Gruppen von Klassen aufgebaut, die sich einem der drei voneinander abhängigen Einheiten Datenmodell, Ansicht und Steuerung zuordnen lassen.

Datenmodell (Model) Darin enthalten sind alle Daten sowie Berechnungsmethoden. Änderungen der Daten werden durch einen Listener erfasst.

Ansicht (View) Ist für die Darstellung des User-Interface auf Basis der Daten des Datenmodells und die Aufzeichnung von Benutzeraktionen durch einen Listener zuständig.

Steuerung (Controller) Implementiert das Listener-Interface und führt den Benutzeraktionen entsprechende Handlungen aus, die das Datenmodell und/oder die Ansicht betreffen und beeinflussen.

7.2.3 Programm und Klassen

Das Programm des Java Applets umfasst mehrere Klassen, in denen logisch zusammengehörende Methoden und Daten, zusammengefasst sind. Die Hauptklasse enthält neben den, für den Zyklus des Applets wichtigen Funktionen, auch eine main Funktion, womit das Applet auch wie ein gewöhnliches Programm, über das Command Window aufgerufen und ausgeführt werden kann. Als Argumente werden beim Aufruf, die IP-Adresse des XPort und die Port Nummer 10001, für das Öffnen der Socket Verbindung, an main übergeben. Nach dem Erzeugen eines Objekts dieser Klasse und dem Setzen eines Flags, welches anzeigt, dass das Programm nicht als Applet gestartet wurde, werden die Funktionen `init()` und `start()` aufgerufen, die der eigentliche Einstiegspunkt des Applets sind, wäre das Programm vom Browser geladen und in der virtuellen Maschine aufgerufen worden.

In der Routine `start()` wird versucht zum Host, von dem das Applet heruntergeladen wurde, eine Socketverbindung aufzubauen, durch Nachfrage der IP-Adresse, sofern es sich um ein Applet handelt. Wenn die Applikation als normales Programm ausgeführt wird, wird die übergebene IP-Adresse dazu herangezogen. Der Aufbau der Verbindung sowie die Methoden zum Senden und Empfangen an zwei Daten-Streams dieses Sockets sind in einer eigenen Klasse festgelegt und deklariert, von der in der Hauptklasse zum Verbindungsaufbau eine Instanz erzeugt wird. Ist eine Verbindung erfolgreich erstellt, wird dies dem Mikrocontroller mitgeteilt. Anschließend folgt die Generierung des grafischen User-Interfaces durch Aufruf des Konstruktors der instanziierten Klasse, die die Ansicht definiert. Die `destroy()` Methode, die vom Browser oder Applet Viewer beim Schließen aufgerufen wird, beendet den Lebenszyklus des Applet und teilt dem Mikrocontroller mit, dass die Verbindung beendet wird sowie stoppt gegebenenfalls auch die Datenübertragung des Mikrocontroller an das Java Applet. Dann erfolgt die Trennung der momentanen Verbindung und das Entfernen des Objektes, welches dafür zuständig ist.

Interaktive GUI Elemente werden mit einem Listener versehen, der die Benutzeraktionen erfasst. Die Klasse, die die Steuerung regelt, implementiert das Listener-Interface und legt die passenden auszuführenden Aktionen fest. Eine weitere Klasse, die die Daten und Berechnungsmethoden umfasst, organisiert die am Input-Stream ankommenden Daten. Durch periodische Ausführung eines Threads werden diese ausgelesen und zur Auswertung in eine dynamische Queue überführt, die ihre Größe den Daten anpasst. Handelt es sich um ein gültiges Datenpaket werden die darin enthaltenen Informationen entsprechend verarbeitet. Messwerte, die vom Datentyp float sind und deren Bytes einzeln übertragen wurden, werden wieder zusammengesetzt und zur Darstellung an das GUI übergeben sowie in einem Array zwischengelagert, welches den Datensatz für die Erstellung des Diagramms bildet. Aus diesem Datensatz holt ein Thread in periodischen Abständen den aktuellen

Wert der momentanen dargestellten Messgröße und fügt in als Folgewert dem Diagramm hinzu. Zur laufenden Darstellung der Messdaten in einem Diagramm wurden Klassen des Graph Package von Leigh Brookshaw verwendet, welches unter GNU General Public Lizenz steht und wesentliche Methoden für eine grafische Erstellung von Funktionen bietet. Die Abbildung 7.6 zeigt die grafische Oberfläche des Applet und die Darstellung der Messwerte als Diagramm, welches sich standardmäßig automatisch skaliert.

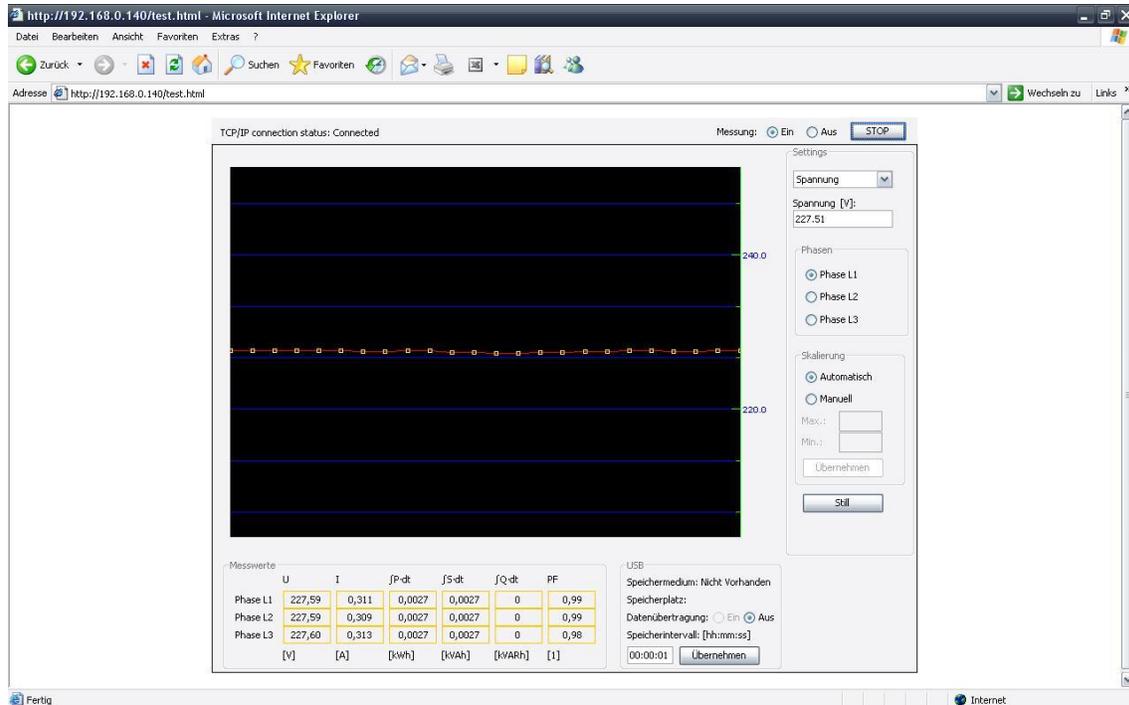


Abbildung 7.6: GUI Java Applet

7.3 Anwendungsspezifisches Datenübertragungsprotokoll

Die Erstellung eines Protokolls zur Kommunikation über die serielle Schnittstelle, die das physikalische Interface zwischen dem XPort und dem Mikrocontroller ist, hängt stark von dem erwarteten Kommunikationsablauf und der auszutauschenden Information ab. Das XPort leitet von den Gesamtdaten eines Ethernet-Frames, der über eine offene Socket-Verbindung eingehen, die Anwendungsdaten direkt an die serielle Schnittstelle weiter. In umgekehrter Reihenfolge werden die, an der seriellen Schnittstelle des XPort ankommenden Daten für den Transport über ein IP Netzwerk in die entsprechenden Protokolle eingekapselt.

Um eine Auslastung des Empfängers, und in weiter Folge verloren gegangene Bytes zu vermeiden, ist ein Handshaking zwischen den Teilnehmern erforderlich, das dem Sender mitteilt ob eine weiter Übertragung zulässig ist. Dies kann hardwaremäßig über zwei zusätzliche Verbindungsleitungen, RTS (Request to Send) und CTS (Clear to Send) erfolgen. Wenn der Empfänger bereit zur Datenaufnahme ist signalisiert er dies über seinen RTS Ausgang, womit dem Sender an seinem CTS Eingang eine mögliche Übertragung

bestätigt wird. Diese Art der Datenflusskontrolle kann mit dem XPort realisiert werden, indem man zwei der allgemeinen GPIO Pins dieser Aufgabe zuweist. Diese Funktion ist am XPort bereits implementiert und kann über die Konfigurationseinstellungen festgelegt werden. Dabei ist zu beachten, dass die Datenflusskontrolle dann zwischen Mikrocontroller und XPort stattfindet und somit nicht direkt zwischen den Endpunkten der Kommunikation, dem Anwendungsprogramm und dem Mikrocontroller. Kontrolliert man die GPIO Pins durch die Anwendungssoftware, die eine Verbindung an Port 30704 aufbaut, können mit einem einfachen Protokoll die Pegel der Pins gesetzt und abgefragt werden. Jede Zustandsänderung oder Abfrage erfordert eine TCP Kommunikation und hat daher eine lange Reaktionszeit.

Eine weitere Möglichkeit zur Realisierung der Datenflusskontrolle besteht mittels Software. Dabei werden zwei Bytes als Kontrollzeichen XON und XOFF verwendet, um den Sender mitzuteilen, dass der Empfänger bereit oder ausgelastet ist. Der verfügbare Zeichenvorrat zur Datenübertragung verringert sich in diesem Fall um die zwei charakteristischen Bytes. Sind aber bei der Datenübertragung alle denkbaren Bitkombinationen möglich, würde das Auftreten der angesprochenen Bytes einen Fehler verursachen.

Die Kommunikationsstruktur zwischen XPort bzw. Java Applet und Mikrocontroller stellt sich wie in Abbildung 7.7 ersichtlich dar.

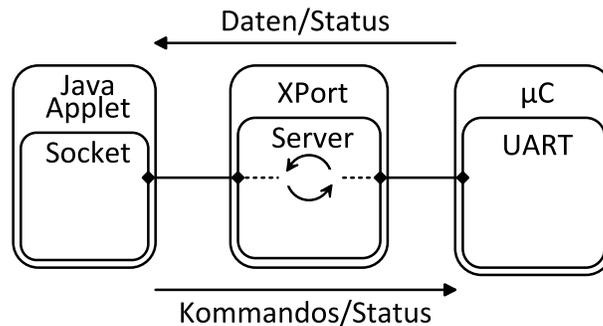


Abbildung 7.7: Kommunikationsstruktur XPort - Java Applet

Der Datenaustausch mit dem XPort erfolgt über die serielle Schnittstelle und ist bidirektional, da sowohl Daten über den Server an das Java Applet weitergeleitet, als auch von diesem empfangen werden. Die Daten, die vom Mikrocontroller gesendet werden, werden dabei in einem Frame nach einem bestimmten Muster zusammengefasst, sodass mit möglichst wenig Bytes so viel Nutzinformation wie möglich übertragen werden kann. So bietet sich auch an, die ermittelten Messwerte, die vom Datentyp float sind, direkt in deren einzelnen Bytes zerlegt zu übermitteln. Zu diesem Zweck wird eine Union definiert, deren verschiedenen Variablen der gleiche Speicherbereich zugeordnet ist. So kann sich ein 4 Byte großes Array vom Typ char den Speicherplatz mit einer float Variable teilen, woraufhin die einzelnen Bytes einer float Variable direkt zugänglich werden. Im Java Applet werden die Daten anhand der bekannten Struktur des Frames wieder rekonstruiert.

Das Java Applet organisiert die an dem Socket empfangenen Bytes über einen Input-Stream, der diese sammelt und aufreihet, bis ein periodisch wiederkehrender Thread sie in ihrer momentanen Gesamtheit ausliest und einem Array übergibt. Um die erhaltenen Informationsdaten zu klassifizieren müssen alle logisch verbundenen Daten vom Sender in Frames organisiert werden, deren Struktur dem Empfänger bekannt ist. Die Bytes des Array

werden in einer dynamischen Queue zwischengelagert wodurch sich eine Fragmentierung der gesendeten Bytes, bedingt durch Programmsprünge im Mikrocontroller, ausgleicht. Die Queue wächst und schrumpft mit der Anzahl der beinhaltenden Daten. Anschließend wird versucht die empfangenen Bytes als gültigen Frame zu verifizieren. Handelt es sich um ein gültiges Datenpaket wird dieses aus der Queue genommen und steht zur weitem Verarbeitung bereit, ansonst wird es verworfen.

Die Struktur der Daten, die an das XPort bzw. Java Applet gesendet werden, ist in Abbildung 7.8 dargestellt.



Abbildung 7.8: Datenframe

Als Startsignal für einen Frame werden zwei Bytes vom Wert 0xFF gesendet. Dann folgt ein Byte als Längeninformation, das die Anzahl aller zu sendenden Bytes wiedergibt, angefangen vom Ersten, dem Startsignal bis zum Letzten, dem Stoppsignal vom Wert 0xFE. Durch die Wahl von einem Byte als Träger der Information bezüglich der Länge, ist die gesamte Framegröße auf maximal 255 Bytes limitiert. Mit Hilfe der an das Applet übermittelten, zu erwartenden Anzahl an Bytes, wird der Frame durch das tatsächliche Auftreten des Stoppzeichens an der damit prognostizierten Position validiert. Sollte ein Frame zerstückelt sein, werden im Extremfall maximal 255 Bytes falsch empfangen. Nun wird in dieser Datenmenge nach einem weiteren Startsignal gesucht, bevor die Daten verworfen werden. Es ist auch möglich bei der Übertragung von mehreren Bytes zusätzlich eine Prüfsumme vor dem Stoppsignal einzufügen, um zusätzliche Sicherheit zu bekommen, dass es sich um einen gültigen Frame handelt. Nach dem Byte für Längeninformation folgt vor den eigentlichen Daten noch ein Byte, welches Auskunft über den eigentlichen Typ des nachfolgenden Datenblocks gibt. An Hand dessen kann das Java Applet entscheiden um welche Art von Daten es sich handelt und wie diese weiter verwertet werden sollen. Neben Messdaten erhält das Applet vom Mikrocontroller zusätzlich Statusinformationen, die die momentanen Betriebszustände des am Mikrocontroller exekutierten, aber auch vom Anwender gesteuerten Programms wiedergeben. Diese Zustände müssen durch eine Übertragung immer aktualisiert werden, falls sich einer oder mehrere durch eine Benutzereingabe am Gerät ändern.

Darüber hinaus hat der Benutzer durch Änderungen der Zustände im Java Applet ebenfalls die Möglichkeit auf den Programmfluss Einfluss zu nehmen und das Gerät so zu steuern. Hierfür sendet das Java Applet nach der Änderung eines Zustandes, durch Drücken oder Auswahl einer Buttons im GUI, eine Datenfolge mehrerer Bytes, die das XPort nach dem Empfang, an die seriellen Schnittstelle weitergibt.

Die UART des Mikrocontrollers wird an die Konfiguration der seriellen Schnittstelle des XPort angepasst. Standardmäßig beträgt die Übertragungsgeschwindigkeit 9600 Baud, doch sind auch höhere Geschwindigkeiten wie beispielsweise 19200 Baud denkbar, was die Übertragungszeit großer Datenframes erheblich reduziert. Der Mikrocontroller XC167 bietet zudem die Möglichkeit den Receive Buffer der seriellen Schnittstelle um einen integrierten FIFO Buffer mit maximal 8 zusätzlichen Speicherplätzen zu erweitern. Dies minimiert die Chance, dass empfangene Daten überschrieben werden noch bevor sie

vom Programm ausgelesen und verarbeitet werden können. Ein Interrupt, der durch jedes empfangene Byte ausgelöst wird, veranlasst das Programm in die Interrupt Service Routine zu verzweigen, die lediglich über Pointer, die im Receive Buffer enthaltenen Werte an einen Ringbuffer weitergibt und somit schnell ausgeführt wird. Dessen Elemente werden über einen Anfangs-Pointer, der das nächste freie Speicherelement anzeigt und einen End-Pointer, der das erste zu lesende Speichersegment markiert verwaltet. Durch regelmäßiges Abfragen diese Ringbuffers in der Hauptschleife werden die vom XPort gesendeten Datenbytes verarbeitet. Um das Protokoll flexibel und tolerant zu gestalten und um eine Art Datenflusskontrolle an der asynchron arbeitenden Schnittstelle zu implementieren, werden die empfangenen Datenbytes durch eine State Maschine verarbeitet, deren Eingangsgrößen die Daten selbst sind. Jedes empfangene Byte versucht den momentanen Zustand dieser State Maschine zu verändern. Nach Abschluss einer gültigen Übertragung wird mittels eines indirekten Funktionsaufrufes eine zur getätigten Übertragung passende Aktion ausgeführt und wieder in den Ausgangszustand verzweigt. Eine ungültige bzw. fehlerhafte Übertragung hat ebenfalls einen Übergang in den Ausgangszustand zur Folge, was in diesem Fall einem Abbruch gleichkommt.

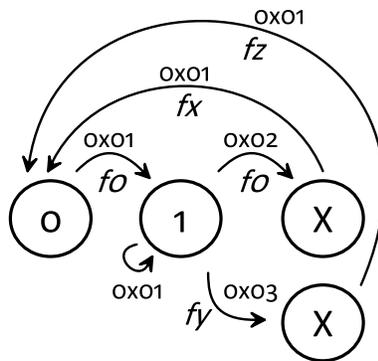


Abbildung 7.9: Ausschnitt Receive State Machine

Die Abbildung 7.9 veranschaulicht den Empfangsvorgang. Durch die Ankunft des Bytes 0x01 springt die State Maschine von Zustand 0 in den Zustand 1, der quasi analog zum besprochenen XON Character eine allgemeine Bereitschaft für den Empfang weiterer Daten repräsentiert. Das nächste übertragene Byte versetzt die State Maschine abhängig von seinem Wert in die Bereitschaft für ein Kommando, welches durch ein folgendes Byte 0x01 als gültig bestätigt und mit einem Funktionsaufruf ausgeführt wird sowie die State Maschine wieder zur Rückkehr zum Zustand 0 veranlasst. Neben diesen vordefinierten Kommandobefehlen lassen sich mit diesem Konzept auch variable Daten bekannter Länge übertragen. Dazu versetzt ein entsprechendes Byte die State Maschine in einen Empfangszustand mit einem gleichzeitigen Funktionsaufruf, der die Veränderung des Zustandes durch ankommende Bytes für die erwartete Datenlänge aussetzt. Die nun einlangenden Daten werden in einem Buffer zwischengespeichert, dessen Inhalt nach der Feischaltung der State Maschine, durch den Erhalt der erwarteten Datenmenge und dem Eintreffen von 0x01 als gültig angesehen und weiterverarbeitet wird.

Kapitel 8

Schlußbemerkung und Ausblick

In vorliegender Diplomarbeit wurde ein Weg zur Realisierung eines Messgerätes zur Erfassung relevanter Parameter im Stromversorgungsnetz mit Webinterface gezeigt, im Zusammenhang mit den wichtigsten Aspekten des Entwurfs und ein Prototyp entwickelt, der für die Erfüllung der Spezifikation die nötigen Funktionen implementiert. Das Ergebnis ist ein Gerät, welches die Durchführung der Messaufgaben, die Datenübertragung über das Netzwerk via Embedded Web Server und die Speicherung auf ein USB Flash-Memory im Dateiformat FAT nach der vorgegebenen Spezifikation beherrscht.

Der im Zuge dieser Arbeit entstanden Prototyp versteht sich als Machbarkeitsstudie bzw. Entwicklungsplattform und bietet verständlicherweise Möglichkeiten zu weiteren Verbesserungen der Funktionalität. Da die Fähigkeiten der vorliegenden Hardware in Verbindung mit der Software noch nicht ausgeschöpft sind, ist es naheliegend vornehmlich Modifikationen im Programm vorzunehmen. Neben allgemeinen Verbesserungen, die die Usability des Mikrocontrollerprogramms und des Java Applets betreffen, besteht auch die Möglichkeit der generellen Erweiterung der Funktionalität. Denkbar ist ein eigener Modus zur Erfassung der genauen Kurvenverläufe von Wechselspannung und Wechselstrom, wie sie an den Eingängen vorliegen und eine direkte Speicherung der ermittelten Daten auf einem USB Flash-Memory zur späteren Auswertung am Computer. Ferner wäre die Bereitstellung einer DC-Betriebsart im vorgegebenen Messbereich vorstellbar, was eine gesonderte Kalibrierung erfordern würde. Potenzial zum Ausbau bietet sich auch im Zusammenhang mit dem Server. Eine Passwortabfrage bei eingehenden Verbindungen wäre im Bezug auf die Sicherheit sinnvoll, ebenso wie eine Verschlüsselung entweder durch das implementierte Rijndael Kryptosystem, welches an dem Port, über den der Datenkanal zwischen Serieller- und Ethernet-Schnittstelle gebildet wird, anwendbar ist und vom Java Applet entschlüsselt werden muss oder durch Einsatz eines anderen XPort Derivats, welches das SSL Verschlüsselungsprotokoll unterstützt. Zur bessern Übersicht der Dateien, die am USB Flash-Memory erstellt werden, empfiehlt sich die Erweiterung der Software um eine Kalenderfunktion, welche diese mit dem Datum und der Uhrzeit des Erstellungszeitpunktes versieht.

Anhang A

Schaltpläne, Layout und Aufbau

Im Anhang finden sich die erstellten Schaltpläne einschließlich des zugehörigen Layouts der einzelnen Platinen, welche zusammen das Gesamtsystem bilden. Ferner sind einige Fotos des entwickelten Prototyps mit offenem Gehäusedeckel beigelegt, die den zusammengesetzten Aufbau abbilden. Außerdem sind noch zwei Tabellen zur Bestimmung von geeigneten Kriech- und Luftstecken angefügt, die der Sicherheitsnorm EN 60950 entnommen sind (vgl. [Hom03]).

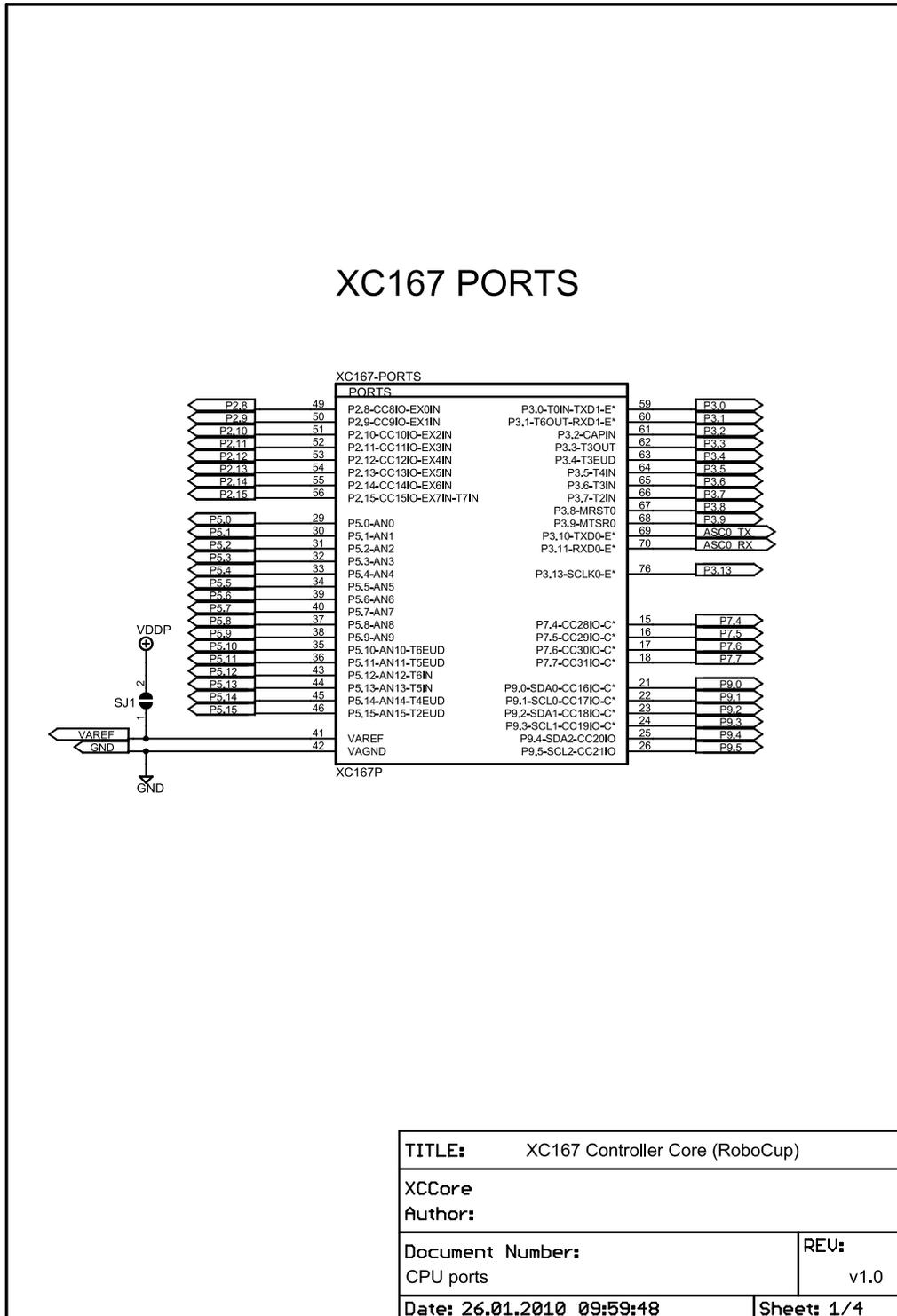


Abbildung A.1: Schaltplan XC167 Kern (vgl.[Kra07])

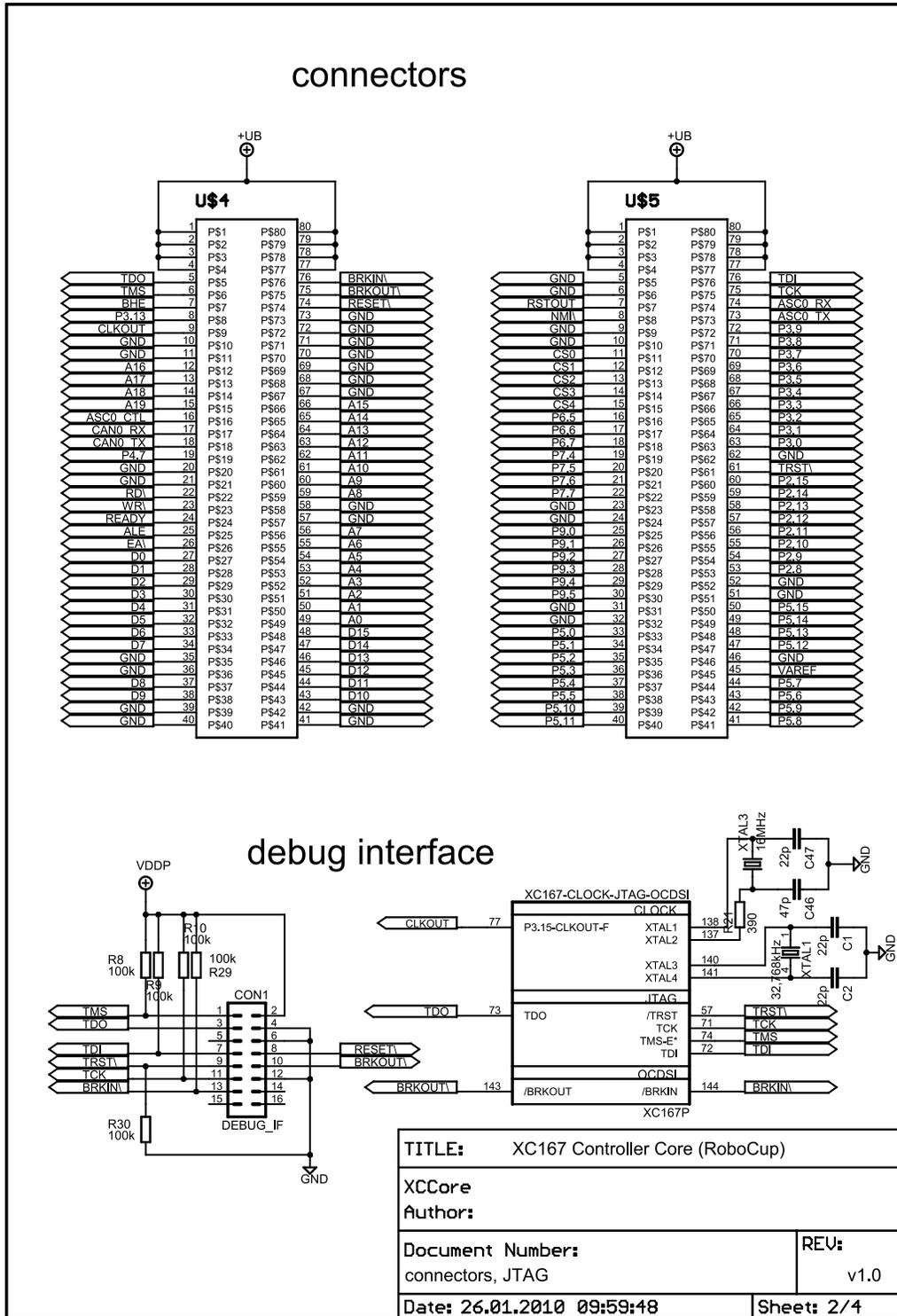


Abbildung A.2: Schaltplan XC167 Kern (vgl.[Kra07])

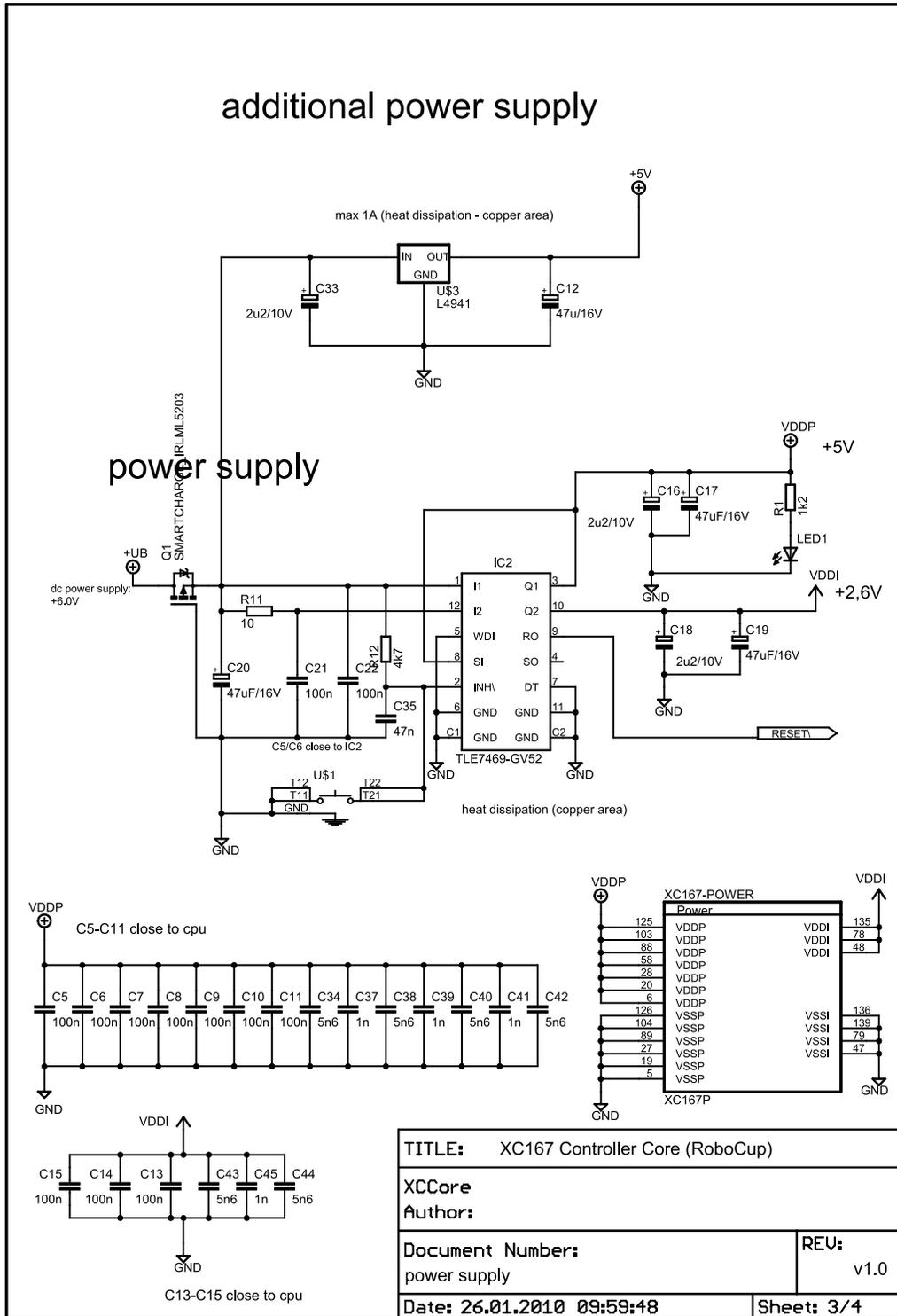
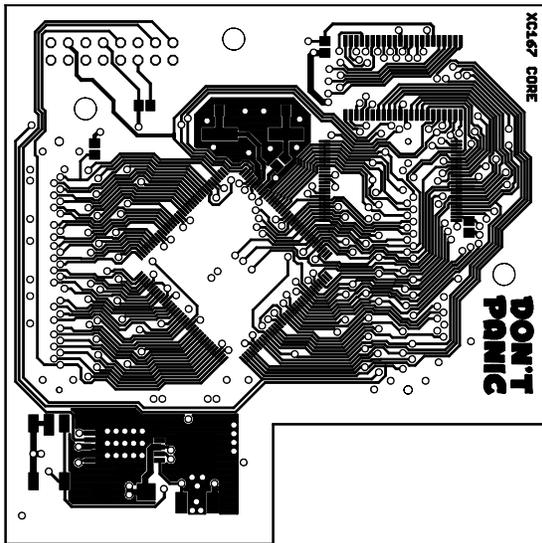
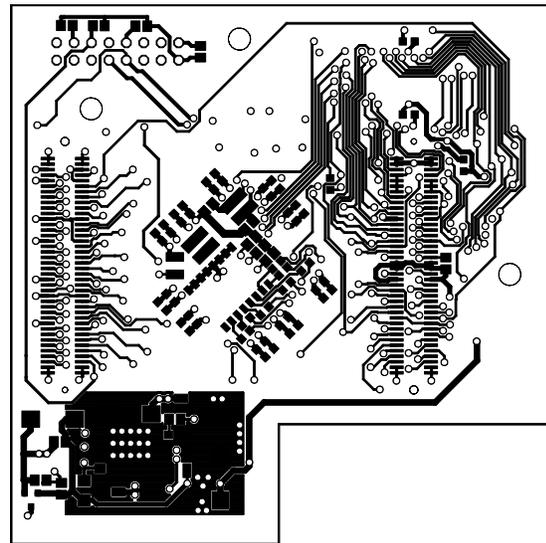


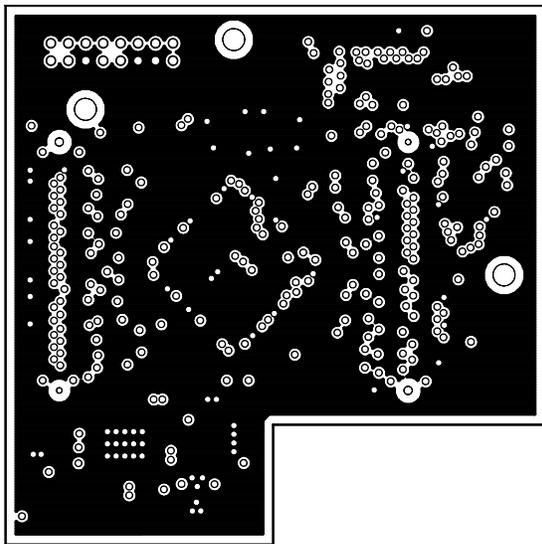
Abbildung A.3: Schaltplan XC167 Kern (vgl.[Kra07])



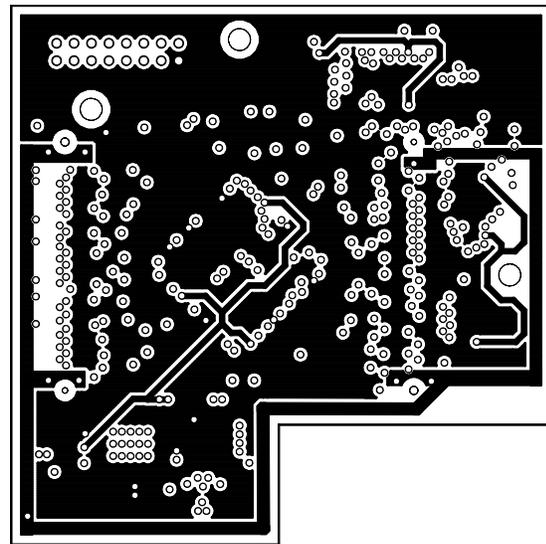
(a) Top



(b) Bottom



(c) Groundlayer



(d) Supplylayer

Abbildung A.5: Layout XC167 Kern (vgl.[Kra07])

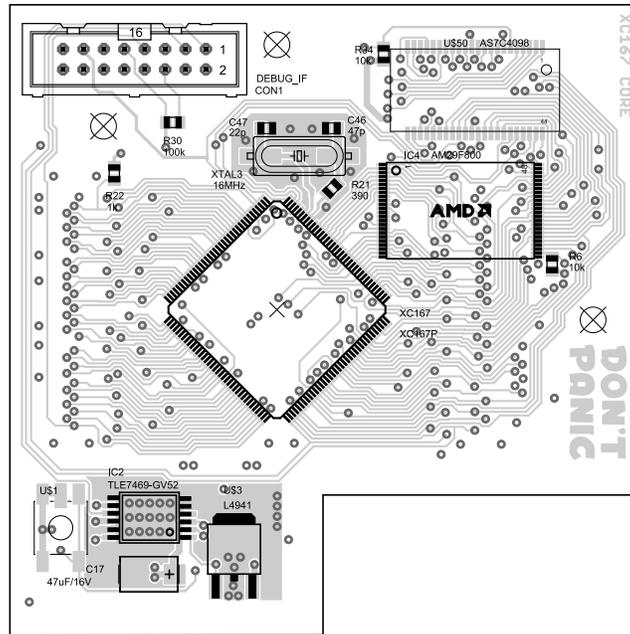


Abbildung A.6: Bestückung Top

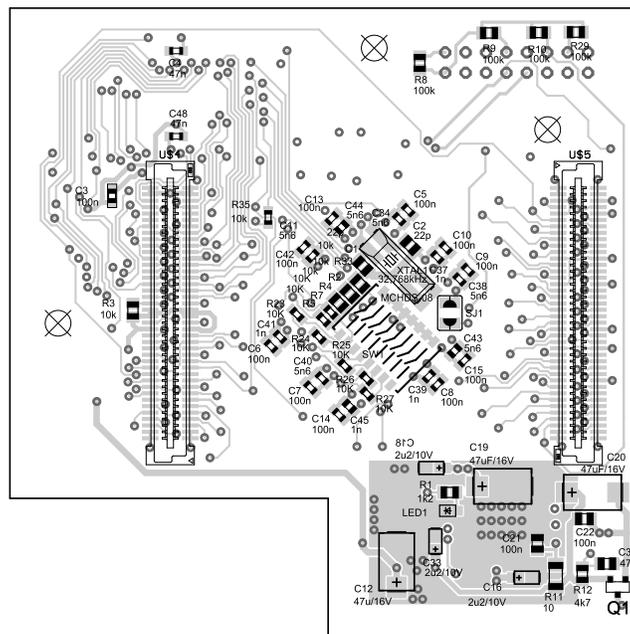


Abbildung A.7: Bestückung Bottom

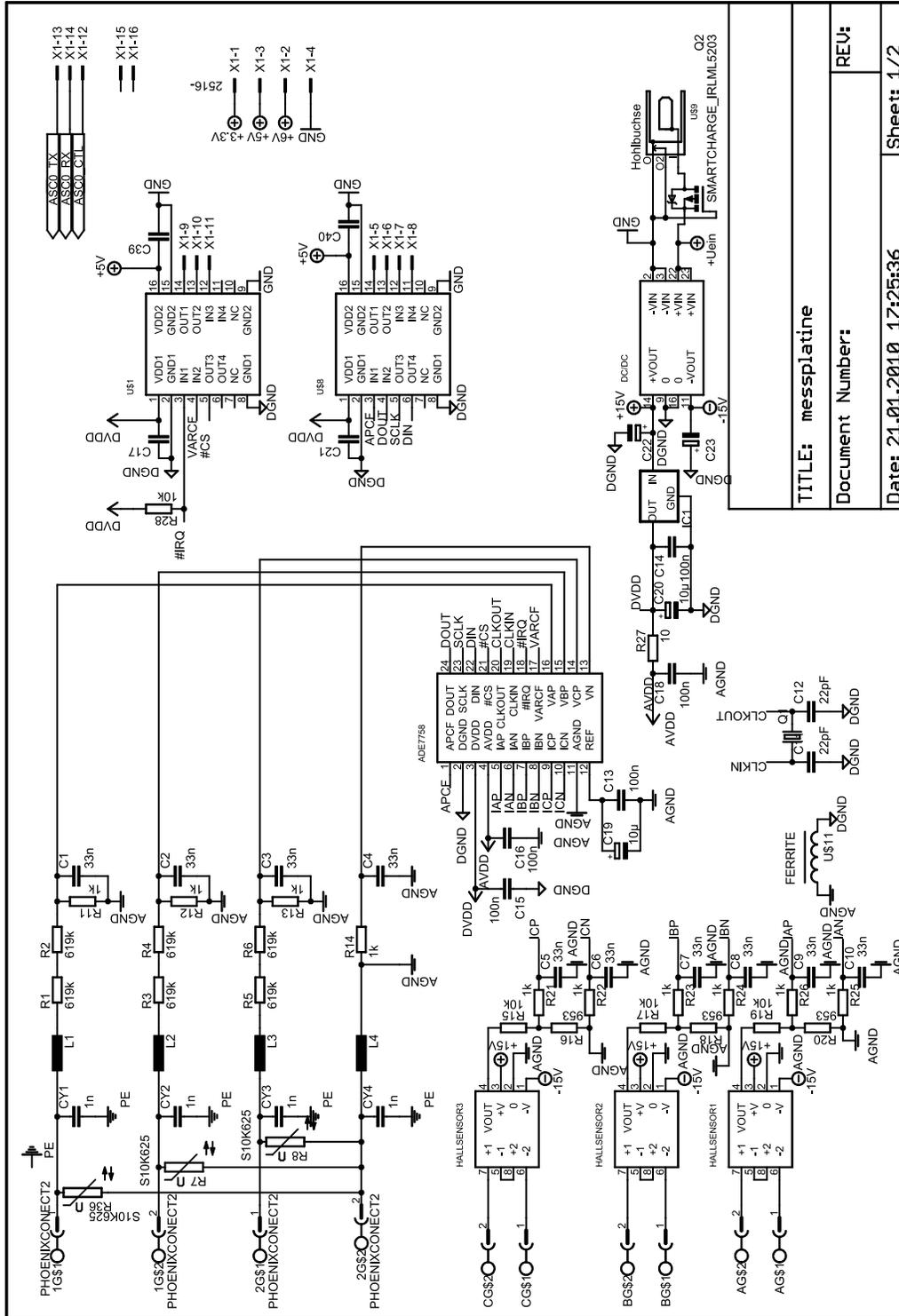


Abbildung A.8: Schaltplan Messplatine

TITLE: messplatine
Document Number:
Date: 21.01.2010 17:25:36
REV: 1
Sheet: 1/2

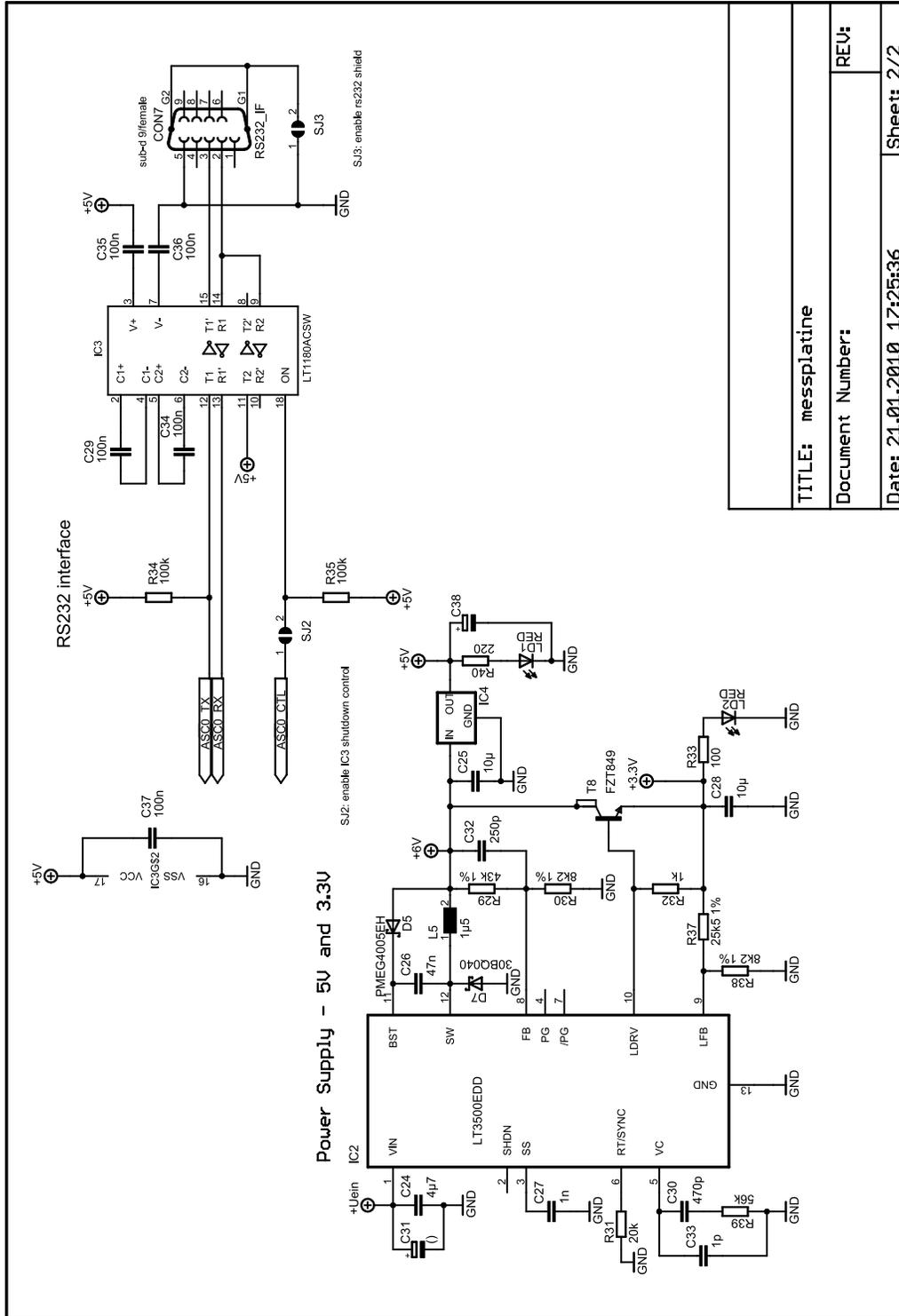


Abbildung A.9: Schaltplan Messplatine

TITLE: messplatine	REV:
Document Number:	
Date: 21.01.2010 17:25:36	Sheet: 2/2

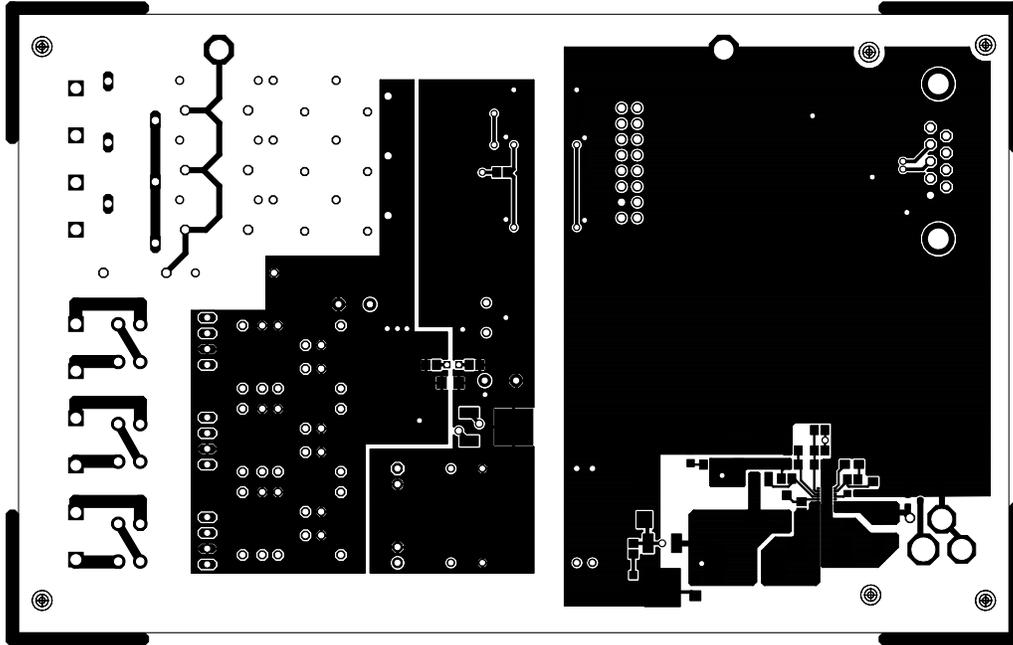


Abbildung A.10: Layout Messplatine TOP

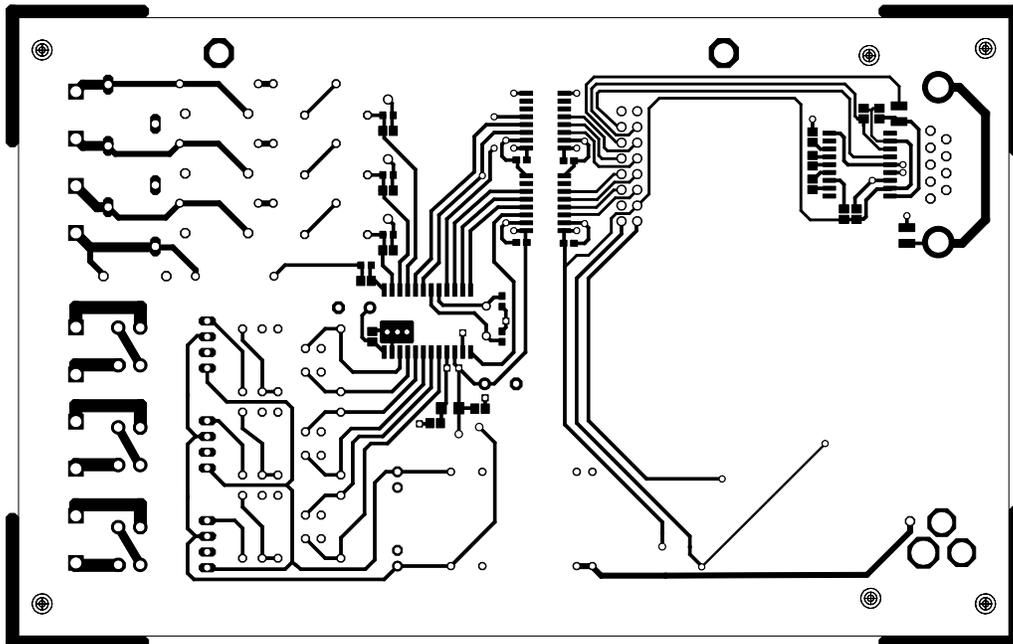


Abbildung A.11: Layout Messplatine BOTTOM

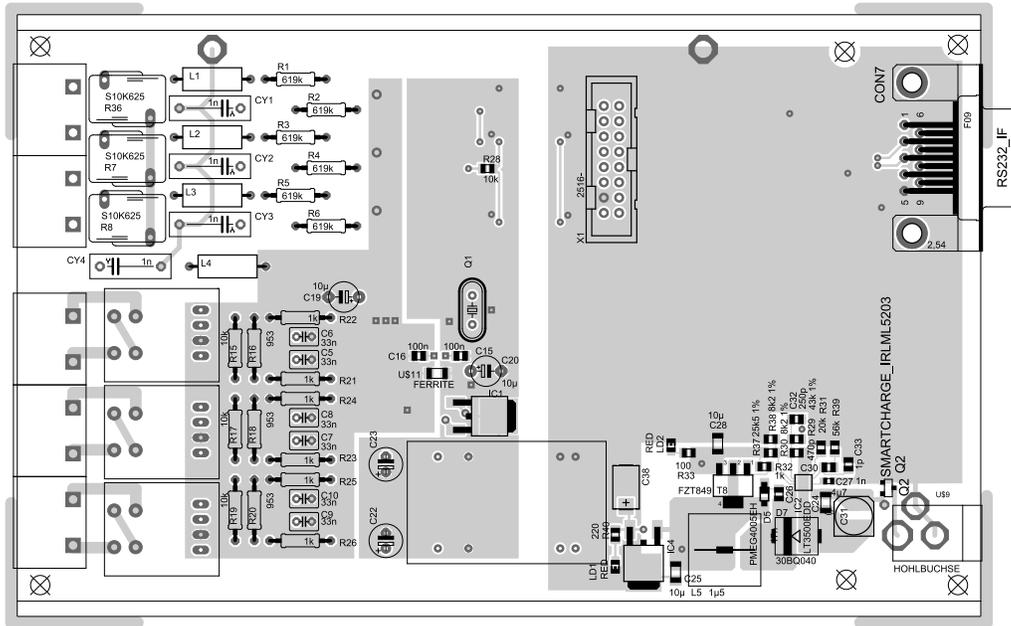


Abbildung A.12: Bestückung Messplatine TOP

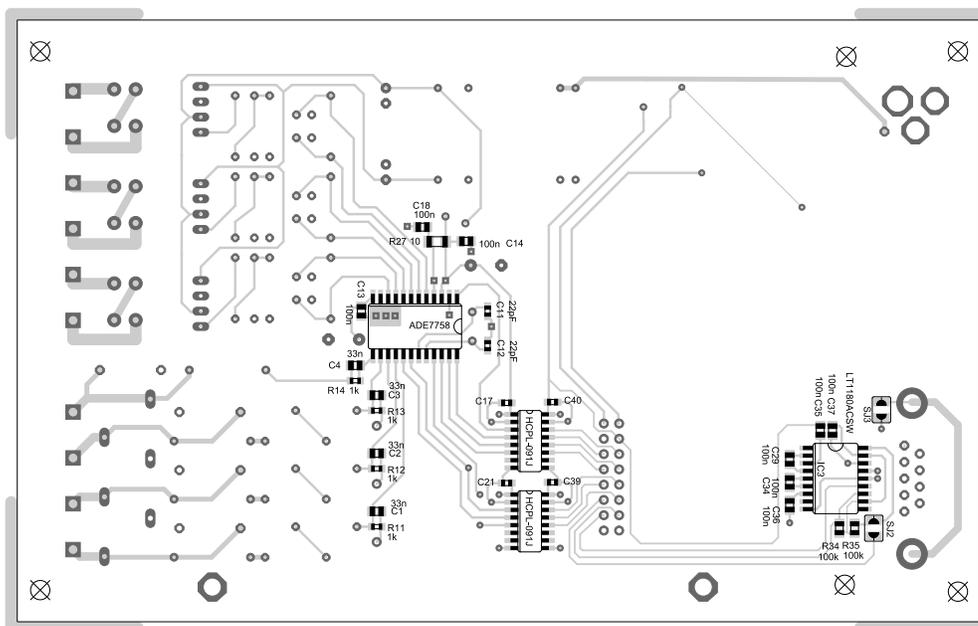


Abbildung A.13: Bestückung Messplatine BOTTOM

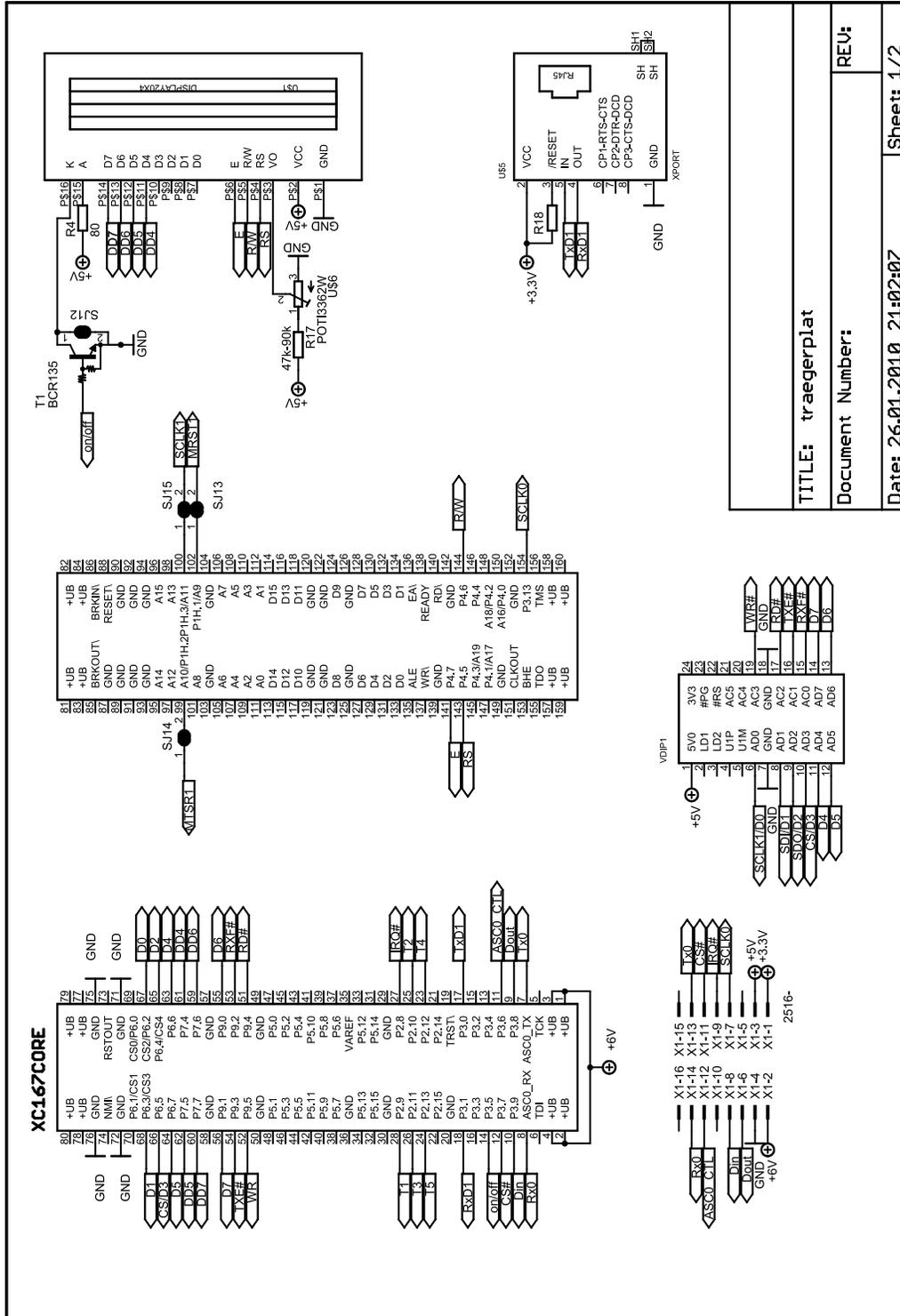


Abbildung A.14: Schaltplan Trägerplatine

TITLE: traegerplat
 Document Number:
 Date: 26.01.2010 21:02:07
 Sheet: 1/2

REV:

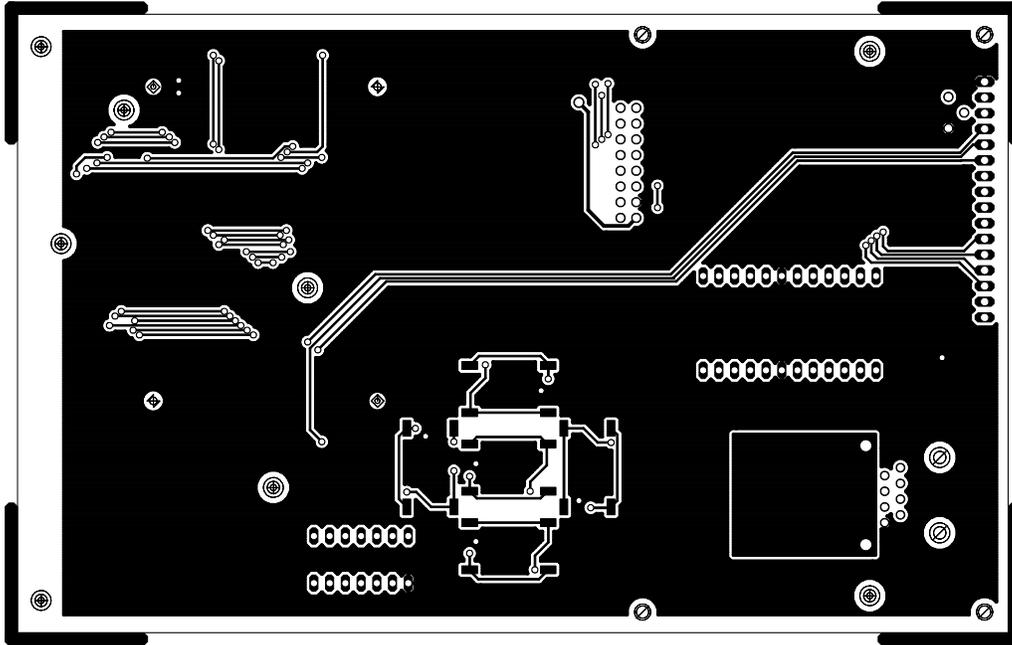


Abbildung A.16: Layout Traegerplatine TOP

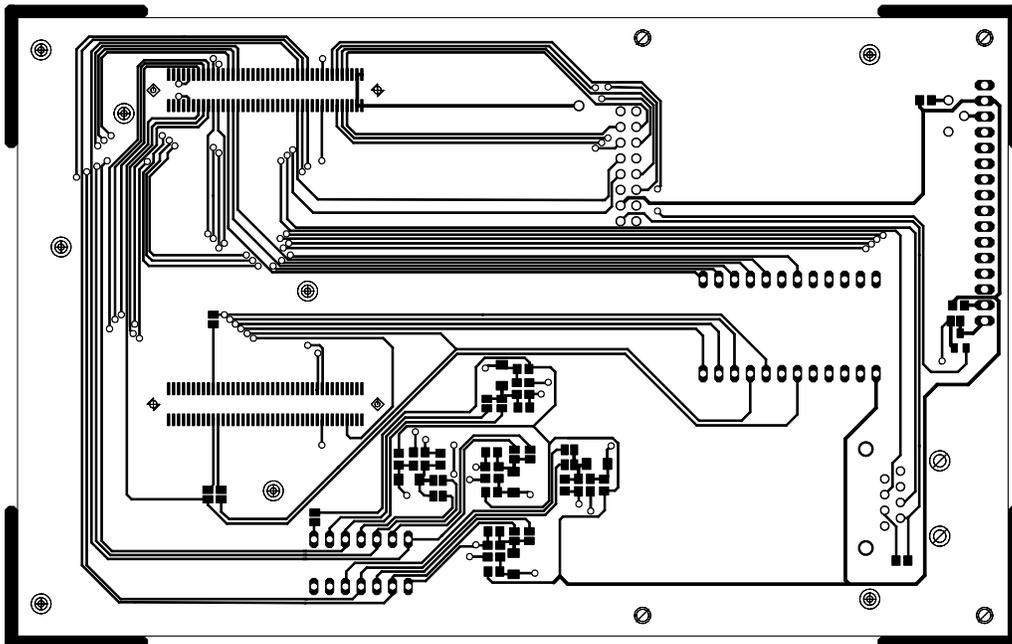


Abbildung A.17: Layout Traegerplatine BOTTOM

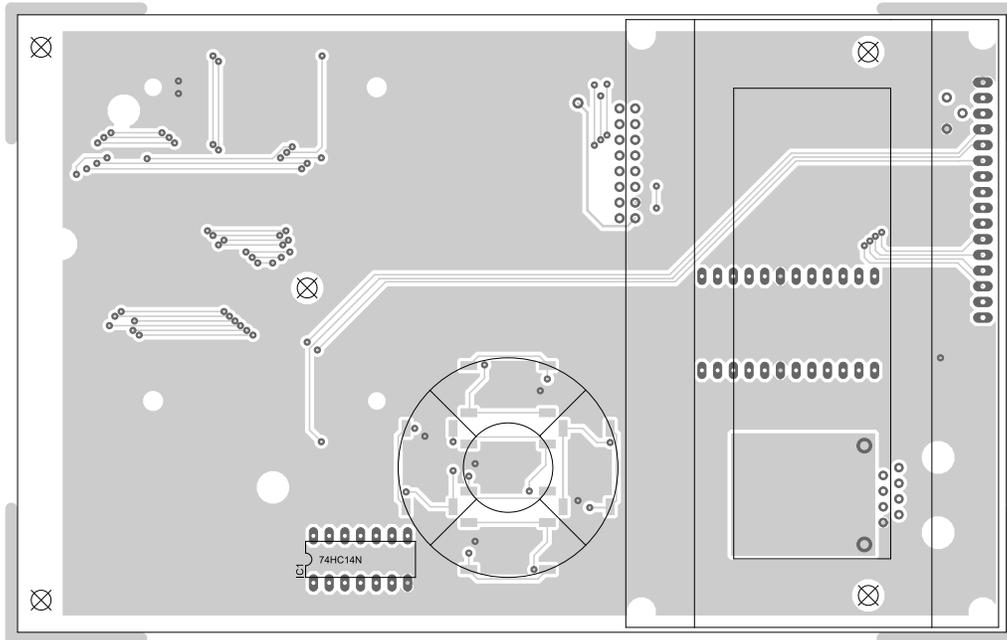


Abbildung A.18: Bestückung Trägerplatine TOP

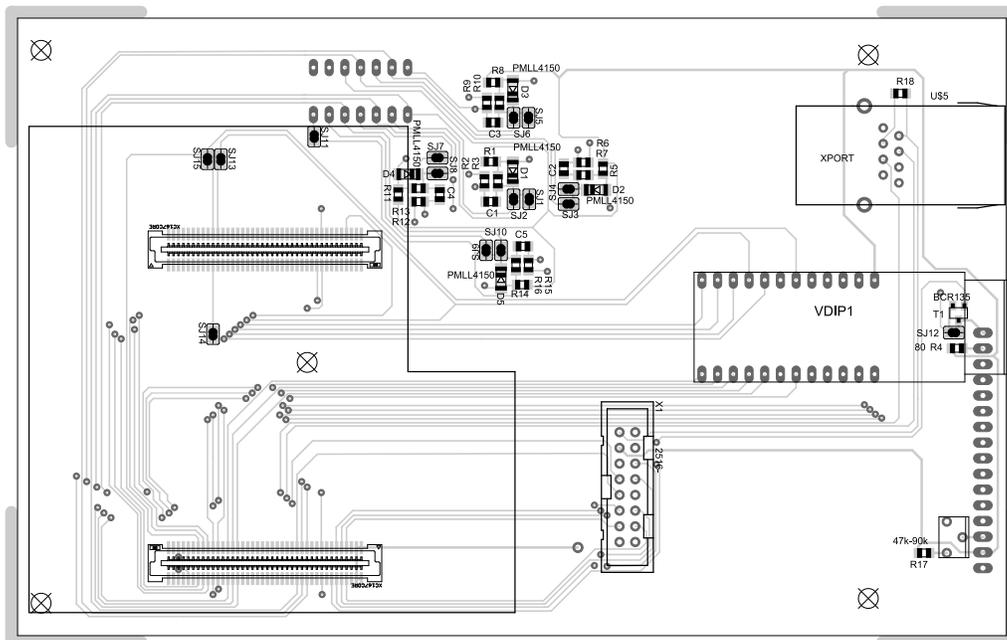


Abbildung A.19: Bestückung Trägerplatine BOTTOM

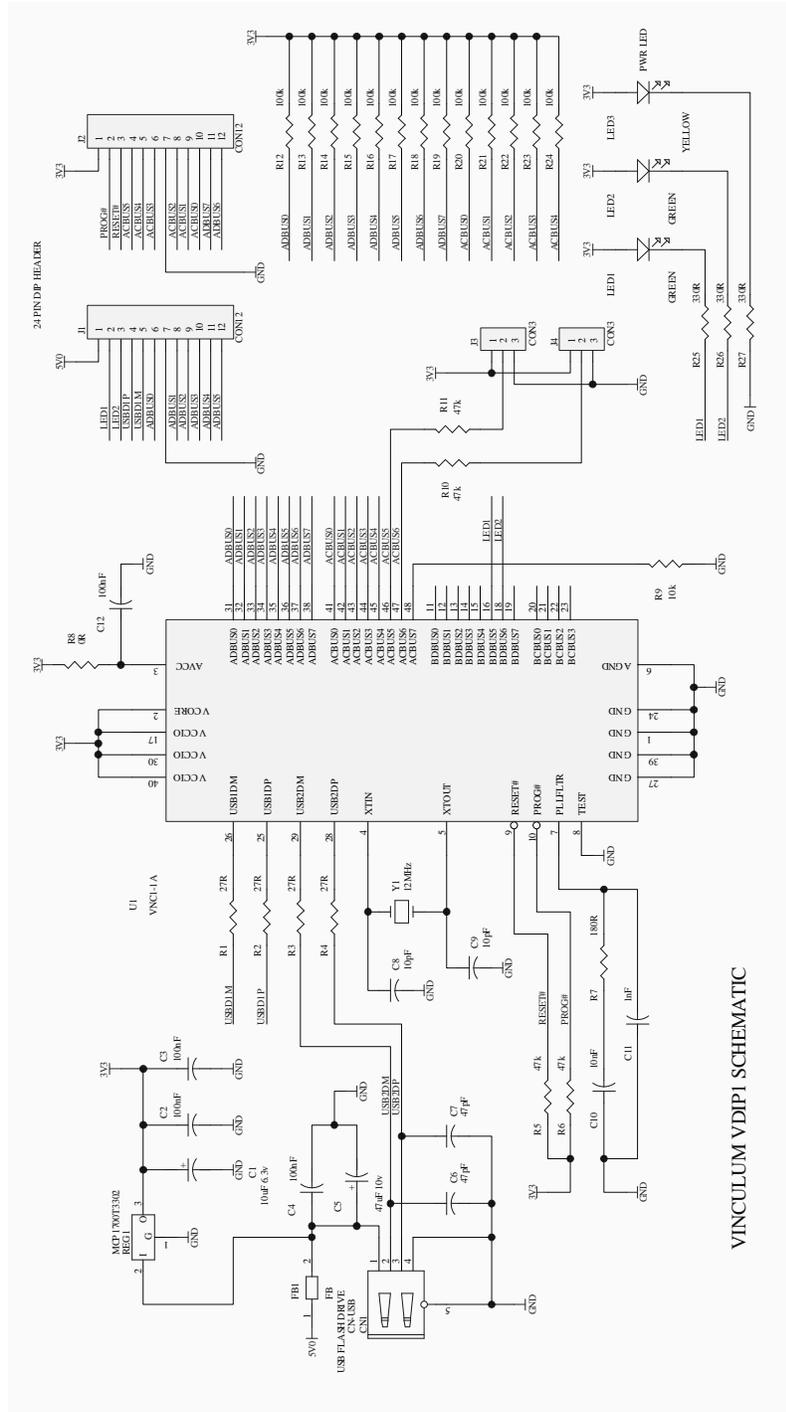


Abbildung A.20: Schaltplan VDIP1 FTDI

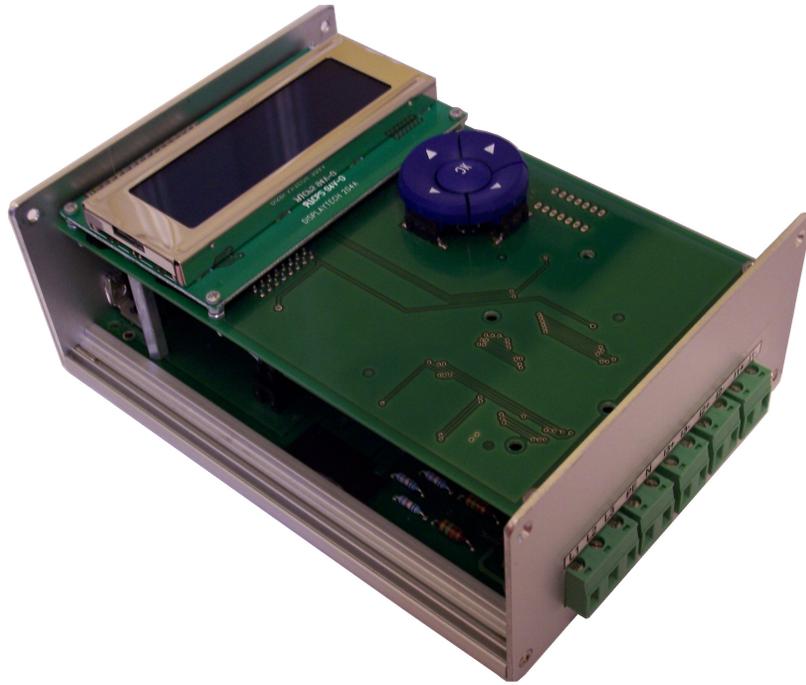


Abbildung A.21: Bild Prototyp

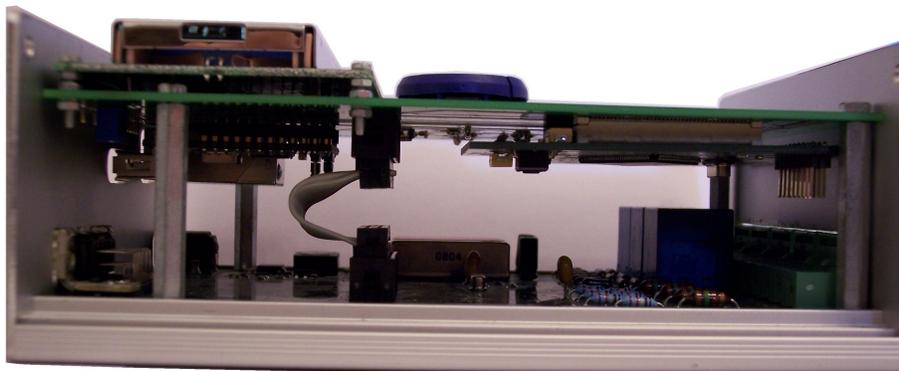


Abbildung A.22: Bild Prototyp

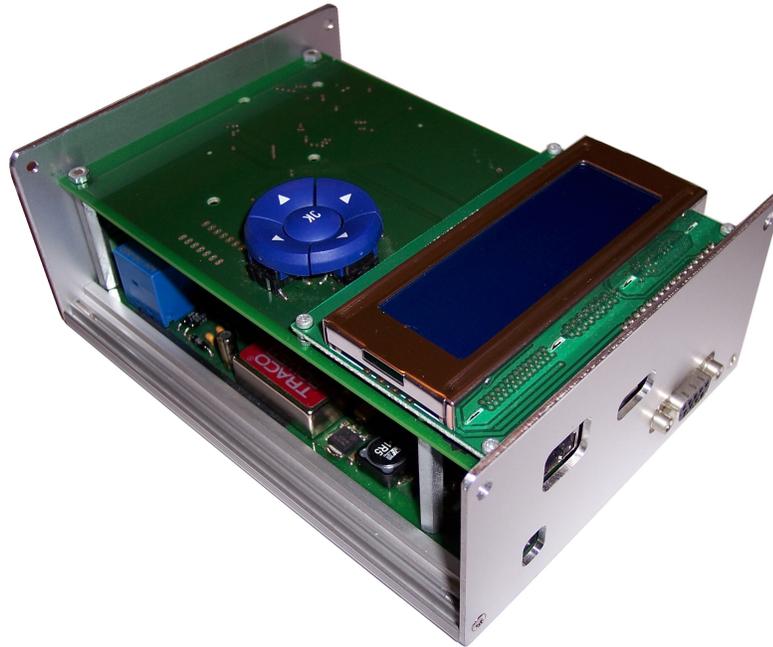


Abbildung A.23: Bild Prototyp

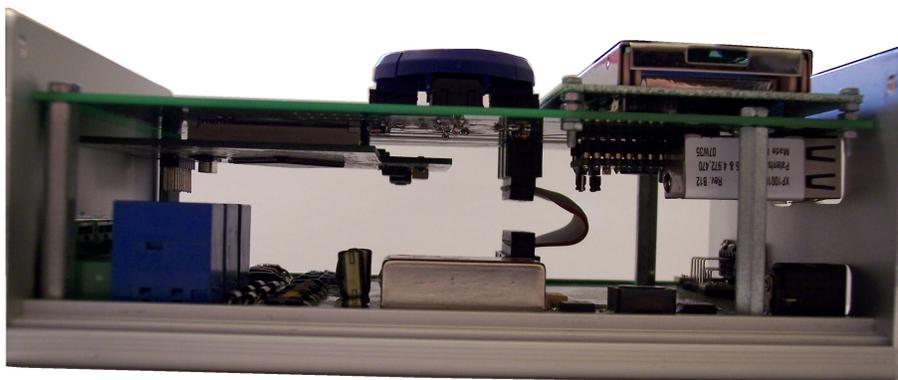


Abbildung A.24: Bild Prototyp

Working Voltage (up to and including)		Nominal Ac Mains Supply Voltage <150V (Mains Transient Voltage 1500V)						Nominal Ac Mains Supply Voltage >150V <300V (Mains Transient Voltage 2500V)						Nominal Ac Mains Supply Voltage >300V <600V (Mains Transient Voltage 4000V)					
Peak or Dc V	Rms (Sinusoidal) V	Pollution Degree 1 and 2			Pollution Degree 3			Pollution Degree 1 and 2			Pollution Degree 3			Pollution Degree 1 and 2					
		F	B/S	R	F	B/S	R	F	B/S	R	F	B/S	R	F	B/S	R			
71	50	0.4	1.0 (0.5)	2.0 (1.0)	0.8	1.3 (0.8)	2.8 (1.6)	1.0	2.0 (1.5)	4.0 (3.0)	1.3	2.0 (1.5)	4.0 (3.0)	2.0	3.2 (3.0)	6.4 (8.0)			
210	150	0.5	1.0 (0.5)	2.0 (1.0)	0.8	1.3 (0.8)	2.6 (1.8)	1.4	2.0 (1.5)	4.0 (3.0)	1.5	2.0 (1.5)	4.0 (3.0)	2.0	3.2 (3.0)	5.4 (6.0)			
420	300	F 1.5 B/S/2.0 (1.5) R 4.0 (3.0)																	
840	600	F 3.0 B/S/3.2 (3.0) R 6.4 (8.0)																	
1400	1000	F/B/S 4.2 R 6.4																	
2800	2000	F/B/S/R 8.4																	
7000	5000	F/B/S/R 17.5																	
9800	7000	F/B/S/R 25																	
14,000	10,000	F/B/S/R 37																	
28,000	20,000	F/B/S/R 80																	
42,000	30,000	F/B/S/R 130																	

1. The values in the table are applicable to functional (F), basic (B), supplementary (S), and reinforced (R) insulation.
 2. The values in parentheses are applicable to basic, supplementary, or reinforced insulation only if manufacturing is subjected to a quality control program that provides at least the same level of assurance as the example given in annex R.2. In particular, double and reinforced insulation shall be subjected to routine tests for electric strength.
 3. For working voltages between 2800 V peak or dc and 42,000 V peak or dc, linear interpolation is permitted between the nearest two points, the calculated spacing being rounded up to the next higher 0.1-mm increment.

Table I. Table 2H of the standard provides minimum clearances for insulation in primary circuits and between primary and secondary circuits (clearances in millimeters).

Abbildung A.25: Luftstrecke EN 60950 [Hom03].

Funtional, Basic, and Supplementary Insulation							
Working Voltage V Rms or Dc	Pollution Degree 1	Pollution Degree 2			Pollution Degree 3		
	Material Group	Material Group			Material Group		
	I, II, IIIa, or IIIb	I	II	IIIa, or IIIb	I	II	IIIa, or IIIb
<50	Use the clearance from the appropriate tables	0.6	0.9	1.2	1.5	1.7	1.9
100		0.7	1.0	1.4	1.8	2.0	2.2
125		0.8	1.1	1.5	1.9	2.1	2.4
150		0.8	1.1	1.6	2.0	2.2	2.5
200		1.0	1.4	2.0	2.5	2.8	3.2
250		1.3	1.8	2.5	3.2	3.8	4.0
300		1.6	2.2	3.2	4.0	4.5	5.0
400		2.0	2.6	4.0	5.0	5.6	6.3
600		3.2	4.5	5.3	8.0	9.5	10.0
800		4.0	5.6	8.0	10.0	11.0	12.5
1000		5.0	7.1	10.0	12.5	14.0	16.0
Linear interpolation is permitted between the nearest two points, the calculated spacing being rounded to the next higher 0.1-mm increment.							
Table IV. Table 2L of the standard provides minimum creepage distances (creepage distances in millimeters).							

Abbildung A.26: Kriechstrecke EN 60950 [Hom03].

Literaturverzeichnis

- [ADE06] Data Sheet ADE7758. *Poly Phase Multifunction Energy Metering IC with Per Phase Information*. ANALOG DEVICES, Norwood, USA, 2006.
- [ADE07] Data Sheet ADE75xx/ADE71xx. *Single-Phase Energy Measurement IC with 8052 MCU, RTC and LCD driver*. ANALOG DEVICES, Norwood, USA, 2007.
- [Col] Anthony Collins. *AN-559 APPLICATION NOTE, A Low Cost Watt-Hour Energy Meter Based on the AD7755*. ANALOG DEVICES, Norwood, USA.
- [FTD06a] FTDI. *VDIP1 Vinculum VNC1L Prototyping Module, Datasheet Version 0.92*. Future Technology Devices International Ltd., Glasgow, UK, 2006.
- [FTD06b] FTDI. *Vinculum VNC1L Embedded USB Host Controller I.C., Datasheet Version 0.97*. Future Technology Devices International Ltd., Glasgow, UK, 2006.
- [FTD08] FTDI. *Vinculum Firmware, User Manual Version 2.05*. Future Technology Devices International Ltd., Glasgow, UK, 2008.
- [Hom03] Ahmadi Homi. Calculating creepage and clearance early avoids design problems later. <http://www.ce-mag.com/>, *Compliance Engineering*, January 2003.
- [Kap03] Rachel Kaplan. *AN-639 APPLICATION NOTE, Frequently Asked Questions (FAQs) Analog Devices Energy (ADE) Products*. ANALOG DEVICES, Norwood, USA, 2003.
- [Kei99] Reinhard Keil. *APPLICATION NOTE 138, In-system FLASH Programming with C16x/ST10*. Keil Elektronik GmbH, München, Germany, 1999.
- [Kra07] Gerald Krammer. Elektronisches Gesamtkonzept für einen mobilen Roboter. Master's thesis, Institute for Electronic, Graz University of Technology, 2007.
- [Lan05] Lantronix. *XPort Data Sheet 910-815E*. Lantronix, Irvine, USA, 2005.
- [Lan08] Lantronix. *XPort Product Brief, Build Network Connectivity into Your Products, Quickly and Simply*. Lantronix, Irvine, USA, 2008.
- [Law] Ken Lawrence. *Safety Fact Sheet 7*. Electrical Safety Consultant, UK.
- [MB06] David Greenhill Michael Beach. *The INSIDER GUIDE to Planning XC166 Family Designs An Engineers Introduction to the XC166 Family*. Infineon Technologies AG, München, Germany, 2006.

- [Sup05] Gerd Suppan. Redesign eines XC167 Mikrocontrollerkerns, 2005.
- [V1.04a] Users Manual V1.0. *XC167-32 16-Bit Single-Chip Microcontroller with C166SV2 Core Volume 1 (of 2): System Units*. Infineon Technologies AG, München, Germany, 2004.
- [V1.04b] Users Manual V1.0. *XC167-32 16-Bit Single-Chip Microcontroller with C166SV2 Core Volume 2 (of 2): Peripheral Units*. Infineon Technologies AG, München, Germany, 2004.
- [V2.03] Application Note AP16031 V2.01. *XC16x Family Migration from C161/C164/C167 Microcontrollers to XC161/XC164/XC167*. Infineon Technologies AG, München, Germany, 2003.
- [Wui03] L. Wuidart. *APPLICATION NOTE, UNDERSTANDING POWER FACTOR*. STMicroelectronics, 2003.